Migrating from TMS320C6455 to TMS320C6474

Jon Bradley

ABSTRACT

The TMS320C6455 fixed-point digital signal processor (DSP) and the TMS320C6474 communications infrastructure DSP are two of Texas Instruments' high-performance DSP processors, each offering high-speed DSP processing, large internal memories, a rich set of peripherals, and other support functions useful in a system environment.

This application report describes device considerations for migrating a design based on the TMS320C6455 to one based on the TI TMS320C6474. These two devices have many similarities; they both contain the TMS320C64x+™ DSP central processing unit (CPU) core, and feature a similar mixture of memory and other peripherals. This document describes the details of the considerations of concern for performing this migration.

Note that since this document describes migration from a TMS320C6455 device to a TMS320C6474, familiarity with the TMS320C6455 device and its documentation is assumed.

Also note that all of the documentation for the TMS320C6455 and the TMS320C6474 referenced in this migration guide can be found on the TI website located in the respective product folders. The device product folders are found at the following two web pages:

http://focus.ti.com/docs/prod/folders/print/tms320c6474.html

http://focus.ti.com/docs/prod/folders/print/tms320c6455.html

Contents

List of Figures

List of Tables

Copyright © 2008, Texas Instruments Incorporated

Trademarks

TMS320C64x+, TMS320C64x+, SmartReflex are trademarks of Texas Instruments.
PCI Express is a registered trademark of PCI-SIG.
RapidIO is a registered trademark of RapidIO Trade Association.
All other trademarks are the property of their respective owners.

Copyright © 2008, Texas Instruments Incorporated

1 Basic Feature Comparison

Table 1 shows a comparison of the basic features of the TMS320C6455 and the TMS320C6474. The remainder of this document presents a comparison of these features in greater detail, and also provides references to the appropriate documentation for further information.

Table 1. Basic C6455/C6474 Feature Comparison

Feature		TMS320C6455	TMS320C6474
CPU Core(s)		1 x C64x+	3 x C64x+
Speed		1.2 GHz	1 GHz
Endianness		Big/Little	Big/Little
Memory	Cache: L1P	32K bytes	32K bytes per core
	Cache: L1D	32K bytes	32K bytes per core
	Cache: L2	2M bytes	3M bytes
	ROM	32K bytes	64K bytes
Peripherals	Antenna Interface		1
	Frame Sync Module		1
	VCP2	1	1
	TCP2	1	1
	EMIFA	1 x 64 bits	
	I2C	1	1
	SRIO	4 (1x links)	2 (1x links)
	DDR	DDR2-533 x 32 bits	DDR2-667 x 32 bits
	DMA	EDMA3	EDMA3
	HPI	16/32 bits	
	PCI	v2.3	
	McBSP	2	2
	EMAC	MII, GMII, RMII, RGMII	SGMII
	Timers	2 x 64 bits	6 x 64 bits
	UTOPIA	1	
	GPIO	16	16
	PLL	2	2
	H/W Semaphores		Yes
Emulation	AET (Advanced Event Trigger)	Yes	Yes
	Trace	Yes	Yes
	ETB (Embedded Trace Buffer)	No	Yes
Power Supplies		4	3
Power Management	SmartReflex		Yes
Power Dissipation	Typical	3.3 Watts	6 Watts
Power Performance		2.9 MIPS/mW	4.0 MIPS/mW
Packages	Type	697 pin BGA	561 pin BGA
	Size	24 x 24 mm	23 x 23 mm
Temperature Range		Commercial/Extended	Commercial

2 Overall Device Considerations

Both the TMS320C6455 and the TMS320C6474 utilize the same C64x+ DSP CPU core. The C6455 contains one DSP CPU core, while the C6474 contains three. The following sections discuss basic considerations of these two devices and the DSP CPU cores provided on the two devices.

Copyright © 2008, Texas Instruments Incorporated

2.1 DSP CPU Core Considerations

Both the C6455 and the C6474 utilize the same TMS320C64x+™ DSP CPU core; therefore, code written for the DSP CPU on the C6455 will generally function in the same fashion on the C6474. Since the C6474 contains three DSP CPU cores, while the C6455 contains one, the C6474 can effectively provide three times the processing power of the C6455 in most applications. When executing applications on the three DSP CPU cores on the C6474, however, additional considerations are often needed for such things as shared memory and resources, algorithm partitioning, and interprocessor communications. The C6474 implements several mechanisms that facilitate these functions in order to support executing an application on this multi-core device. These mechanisms are discussed in further detail in later sections of this document.

Note that there are also differences in other aspects of these devices that affect how code operates, such as functionality of peripheral modules, and differences in memory map, and these differences must be accounted for when migrating applications. These additional differences are also discussed in detail in the remainder of this document.

2.2 CPU Clock Speeds

The C6455 and C6474 can be operated at a range of clock speeds to accommodate a variety of different performance requirements. In addition, the C6455 is also available in four different speed versions in order to address different system speed and cost tradeoffs.

Table 2 shows a summary of the different speed versions of the C6455 and C6474.

Table 2. Available Performance Versions of the C6455 and C6474

C6455		C6474
Version	Speed	Speed
6455-2	1.2 GHz (0.83 ns cycle time)	
6455	1 GHz (1.0 ns cycle time)	1 GHz (1.0 ns cycle time)
6455-8	850 MHz (1.17 ns cycle time)	
6455-7	720 MHz (1.39 ns cycle time)	

Note that the CPU speeds listed in Table 2 are the maximum operating speeds for the CPUs in question. Actual operating speeds are frequently chosen to match clocking requirements of the system as a whole. This results in clocks for other parts of the devices to match with requirements for their applications.

Note also that the power supply voltage requirements are different on these devices for some of the different speed versions. See Section 10 of this document for detailed information regarding power supply voltage requirements for the C6455 and C6474. Also, see Section 4.11 of this document for information regarding operation of the PLL/clock generators on these two devices, and Section 8 for information regarding initialization of the PLL/clock generators on these two devices.

For additional detailed information regarding performance, timing requirements, and characteristics for the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552).

2.3 Endianness Considerations

The C6455 and the C6474 both function in either little-endian or big-endian operating mode, therefore, there are no endianness considerations when migrating an application from the C6455 to the C6474.

3 Internal Memory Comparisons

Both the C6455 and the C6474 feature on-chip internal memories, allowing efficient handling of varied partitions of internal program and data information. Both devices feature several different types of cache memory, allowing significant flexibility in using this memory to enhance algorithm performance. Both devices also provide an on-chip ROM, which contains the bootloader program. Since there are some differences between the memory architectures on the two devices, some software modifications may be required when migrating applications from the C6455 to the C6474.

Table 3 shows a comparison of the C6455 and C6474 DSP internal memory.

Copyright © 2008, Texas Instruments Incorporated

Table 3. DSP Internal Memory Comparison

Memory Type	C6455	C6474
L1P Program Memory	32K-Bytes RAM/Cache (direct mapped), flexible allocation	32K-Bytes RAM/Cache (direct mapped), flexible allocation
L1D Data Memory	32K-Bytes RAM/Cache (2-way set associative), flexible allocation	32K-Bytes RAM/Cache (2-way set associative), flexible allocation
L2 RAM Memory	2M-Bytes Unified Mapped RAM/Cache (4-way set associative), flexible allocation	3M-Bytes Unified Mapped RAM/Cache (4-way set associative), flexible allocation
Maximum L2 Cache Size	256K-Bytes	256K-Bytes
ROM	32K-Bytes	64K-Bytes

For additional detailed information regarding use of the C6474's internal memory, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C64x+ DSP Cache User's Guide* (SPRU862). For additional information regarding the bootloader on the C6474, see Section 6 of this document.

4 Peripherals

The TMS320C6455 and TMS320C6474 both feature a wide variety of peripheral modules that are useful in many different system environments. This section presents a comparison of the peripheral offerings on these two devices.

4.1 Antenna Interface (AIF)

The C6474 features the addition of a specialized AIF peripheral that can be used to provide added communications capability in C6474 systems. The primary function of the AIF peripheral is to implement a glueless interface to an RF antenna module that provides multichannel communications capability, which is required in the telecommunications base-station environment. In addition, however, the AIF can be used to implement basic data transfer capabilities either between C6474 DSPs, or with other devices in the system.

External to the C6474, the antenna interface consists of six bi-directional serial channels that can each transfer data at up to 3.072G bits/second. Each of these bidirectional serial channels consists of a transmit and receive differential signal pair, implemented using a serializer/deserializer module similar to that used by Serial RapidIO® (SRIO) or PCI Express® (PCIe) interfaces.

Internal to the C6474, the AIF implements all of the features necessary to meet the requirements of two industry-standard communications protocols: the Open Base Station Architecture Initiative (OBSAI) and the Common Public Radio Interface (CPRI). Compatibility with these industry communications standards is indispensable in meeting the requirements of telecommunications base-station applications. The frame sync module (FSM) companion peripheral to the AIF (see Section 4.3) handles system timing-related tasks such as frame synchronization and enhanced direct memory access (EDMA) event generation.

Additionally, however, although the AIF uses its own unique protocol, since this peripheral provides data transfer capabilities that are comparable to six lanes of SRIO, when appropriately configured, it can provide high-speed general-purpose data transfers that can be used in lieu of SRIO or PCIe. Since the six channels of AIF data can be used together for aggregated data transfers, the overall data throughput rate that can be provided by this interface can be extremely significant.

For detailed information regarding use of the AIF peripheral on the C6474, see the *TMS320C6474 Antenna Interface User's Guide* (SPRUG12).

4.2 Frame Synchronization Module (FSM)

The FSM functions with the AIF to support the timing, frame sync generation, and EDMA event generation required by the AIF. Since both the AIF and FSM are additional modules on the C6474 and not included on the C6455, their usage is unique to the C6474.

Copyright © 2008, Texas Instruments Incorporated

The FSM primarily consists of several sets of timers and counter registers necessary to generate timing signals needed by the AIF, including frame sync, and various EDMA events to properly time data transfers that are required when using the AIF peripheral. The FSM also provides a watchdog timer, a time of day counter, and error and alarm handling.

For proper functioning of the FSM, this peripheral must be correctly initialized in order to generate these timing signals and EDMA events. For additional detailed information regarding the FSM, see the *TMS320C6474 Frame Synchronization User's Guide* (SPRUG13).

4.3 Viterbi Decoder Coprocessor (VCP2)

The C6455 and the C6474 both feature the VCP2 coprocessor which, in conjunction with the turbo-decoder coprocessor 2 (TCP2) coprocessor, can be used to significantly accelerate channel-decoding operations on-chip. The VCP2 operating at CPU clock divided-by-3 can decode over 694 7.95-Kbps adaptive multi-rate (AMR) [K=9, R=1/3] voice channels. The VCP2 supports constraint lengths K = 5, 6, 7, 8, and 9, rates R = 1/2, 1/3, and 1/4, and flexible polynomials, while generating hard decisions or soft decisions. Communications between the VCP2/TCP2 and the CPU are implemented through the EDMA3 controller.

The same VCP2 coprocessor peripheral module is used on both devices; therefore, migration of an application utilizing the VCP2 requires little, if any, modification of software. This is further facilitated by the fact that all of the VCP2 module memory mapped control registers are located at the same addresses in both devices.

For detailed information regarding the use of the VCP2 on the C6474, see the *TMS320C6474 DSP Viterbi-Decoder Coprocessor 2 (VCP2) Reference Guide* (SPRUG20).

4.4 Turbo Decoder Coprocessor (TCP2)

The C6455 and the C6474 both feature the TCP2 coprocessor which, in conjunction with the VCP2 coprocessor, can be used to significantly accelerate channel-decoding operations on-chip. The TCP2 operating at CPU clock divided-by-3 can decode up to fifty 384-Kbps or eight 2-Mbps turbo encoded channels (assuming 6 iterations). The TCP2 implements the Max*-Log-MAP and Max-Log-MAP algorithms and is designed to support all polynomials and rates required by Third-Generation Partnership Projects (3 GPP and 3 GPP2), with fully programmable frame length and turbo interleaver. Decoding parameters such as the number of iterations and stopping criteria are also programmable. Communications between the VCP2/TCP2 and the CPU are implemented through the EDMA3 controller.

The same TCP2 coprocessor peripheral module is used on both devices; therefore, migration of an application utilizing the VCP2 requires little, if any, modification of software. This is further facilitated by the fact that all of the VCP2 module memory mapped control registers are located at the same addresses in both devices.

For detailed information regarding the use of the TCP2 on the C6474, see the *TMS3206474 DSP Turbo-Decoder Coprocessor 2 (TCP2) Reference Guide* (SPRUG21).

4.5 External Memory Interfaces (EMIF)

Both the C6455 and the C6474 feature flexible external interfaces that support accessing various types of memories. The C6455 supports two independent memory interfaces, each architected for specific memory types. These two memory interfaces are the DDR2 memory interface, specifically designed to gluelessly interface to industry-standard DDR2 memories, and the general-purpose parallel external memory interface (EMIFA), designed to efficiently interface to a variety of synchronous and asynchronous memory types. The C6474 provides a high-speed DDR2 memory interface. The following paragraphs describe migration of applications using these two interfaces from the C6455 to the C6474 device.

4.5.1 DDR2 EMIF

On the C6455 and the C6474 devices, a virtually identical peripheral module with a very similar feature set is used for the DDR2 interface, therefore, migration of an application utilizing the DDR2 EMIF between these two devices should require only minimal modifications.

Copyright © 2008, Texas Instruments Incorporated

The DDR2 interface utilizes a 32-bit data bus, and is optimized for use with high-speed, high-density DDR2 memory for storage of programs and large blocks of data. Additionally, both devices also have the capability to use the interface data bus in 16-bit mode, instead of 32-bit mode, if desired, to save pinout connections. Both devices can access a total of 512 Mbytes of DDR2 address space.

Since the C6455 and the C6474 utilize almost identical DDR2 EMIF peripheral modules, the predominant difference in usage between this peripheral module on these two devices is the fact that the C6455 DDR2 interface runs up to 533 MHz, while the C6474 can run up to 667 MHz. Therefore, although the C6474 DDR2 interface can be initialized identically to the C6455 DDR2 interface and run at 533 MHz (when clock frequencies are the same), in order to function at 667 MHz, the C6474 requires clocking and interface parameters to be initialized differently. Note that the proper initialization of the DDR2 interface is unique to the specific type of DDR2 memories being used, therefore, the device-specific data sheet should be consulted. Also, for proper DDR2 operation at 667 MHz, appropriate PC board design and layout guidelines must be followed. For detailed information regarding DDR2 PC board design and layout for the C6474, see the *TMS320C6474 DDR2 Implementation Guidelines* (SPRAAW8).

In addition to differences mentioned above, there are a few other minor differences between the DDR2 EMIF on these two devices. The first of these is that the locations within the device address space at which the DDR2 memory and the DDR2 EMIF control registers are accessed are different. For the C6455, the DDR2 memory is accessed starting at address E0000000h, and the base address of the control registers is 78000000h, whereas on the C6474, the DDR2 memory is accessed starting at address 80000000h, and the base address of the control registers is 70000000h.

Another difference in the DDR2 peripherals on these two devices is that the C6474 has the added capability to support on-die termination (ODT) resistors for the DDR2 memories, while the C6455 does not. The ODT capability on the C6474 is controlled by the DDR2IO register located at address 700000F0h.

The only other difference between the two devices is that the value of the Module and Revision Register (MIDR) contains a different value on each device. This is due to the fact that an updated revision of the peripheral supporting ODT was used on the C6474. Therefore, the C6455 MIDR contains 0031030Fh, while the C6474 MIDR contains 0031031Bh.

A comparison between the DDR2 EMIF on the C6455 and the C6474 is shown in Table 4.

Table 4. Comparison of DDR2 EMIF Features

Feature		C6455	C6474
Data Width		32 or 16 bits	32 or 16 bits
Speed		533 MHz	667 MHz
Base Address in Device Address Space		E0000000h	80000000h
Address Range		512 Mbytes	512 Mbytes
Control Register Base Address		78000000h	70000000h
ODT Supported		No	Yes
ODT Control Register Address	DDR2IO		700000F0h
Register Default Initialization Value	MIDR	0031030Fh	0031031Bh

For additional detailed information regarding use of the DDR2 EMIF on the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552), and the *TMS320C6474 DSP DDR2 Memory Controller User's Guide* (SPRUG19).

4.5.2 Synchronous/Asynchronous External Memory Interface (EMIFA)

The C6455 offers an EMIFA, while the C6474 does not, therefore, for C6474 applications requiring additional external storage, other provisions must be made for this.

Since the C6474 implements a large memory space using DDR2, if significant amounts of memory are required in a particular application, the DDR2 space should be fully populated.

Copyright © 2008, Texas Instruments Incorporated

If additional storage is required beyond the maximum that can be accommodated in DDR2 memory, information can be accessed in an external device using other peripheral interfaces on the device such as the inter-integrated circuit (I2C), multichannel buffered serial port (McBSP), Ethernet media access controller (EMAC) or AIF.

4.6 Enhanced Direct Memory Access (EDMA) Controllers

The C6455 and the C6474 both feature EDMA controllers that can be used to transfer data to and from numerous locations, both on- and off-chip. Both devices support 64 independent channels of EDMA transfers.

The EDMA controllers used on these two devices are both based on the EDMA 3.0 peripheral module, however, there are some differences between the peripheral modules on the two devices. Specifically, the C6455 has four transfer controllers and four queues, while the C6474 has six transfer controllers and six queues. Also, the C6474 EDMA provides the flexibility with its transfer controllers for either 128-bit or 64-bit data bus widths to optimize transfer efficiency, while the C6455 provides only a data bus width of 128 bits.

Also, note that migration from the C6455 to the C6474 is further facilitated by the fact that the memory mapped control registers for the channel controller and the first four transfer controllers on the C6474 are located at the same memory addresses.

Table 5 presents a comparison of the EDMA controllers on the C6455 and the C6474.

Table 5. C6455 to C6474 EDMA Comparison

Memory Type	C6455	C6474
Number of Transfer Controllers	4	6
Number of Queues	4	6
Number of QDMA Channels	4	8
Data Bus Width	128 bits	TC0-2: 128 bits, TC3-5: 64 bits
Default Burst Size	Fixed - 64 bytes	Fixed - 64 bytes
Transfer Completion Interrupts	9 (Global + Shadow Region 0-7)	9 (Global + Shadow Region 0-7)
Error Interrupts	6 (Global, Mem. Protection, TC0-3)	8 (Global, Mem. Protection, TC0-7)

For both the C6455 and the C6474, the 64 possible EDMA channel synchronization events are predefined to various sources on the device, and the actual synchronization events used to trigger specific EDMA operations are selected from these possible synchronization events. Note that on the C6474, additional flexibility in the selection of EDMA channel synchronization events is provided by CIC[3], which further multiplexes system events to be used for EDMA synchronization. Refer to Section 5 for further information regarding the use of the CICs on the C6474. Table 6 presents a comparison of the 64 possible EDMA channel synchronization events available on the C6455 and the C6474.

Table 6. EDMA Channel Synchronization Event Comparison

EDMA Channel	C6455		C6474	
	Event Name	Event Description	Event Name	Event Description
0	DSP_EVT	HPI/PCI-to-DSP event	TINT0L	Timer Interrupt Low
1	TEVTLO0	Timer 0 lower counter event	TINT0H	Timer Interrupt High
2	TEVTHI0	Timer 0 high counter event	TINT1L	Timer Interrupt Low
3	-	None	TINT1H	Timer Interrupt High
4	-	None	TINT2L	Timer Interrupt Low
5	-	None	TINT2H	Timer Interrupt High
6	-	None	CIC3_EVT0	CIC_EVT_o [0] from Chip Interrupt Controller

Copyright © 2008, Texas Instruments Incorporated

Table 6. EDMA Channel Synchronization Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Event Name	Event Description	Event Name	Event Description
7	-	None	CIC3_EVT1	CIC_EVT_o [1] from Chip Interrupt Controller
8	-	None	CIC3_EVT2	CIC_EVT_o [2] from Chip Interrupt Controller
9	-	None	CIC3_EVT3	CIC_EVT_o [3] from Chip Interrupt Controller
10	-	None	CIC3_EVT4	CIC_EVT_o [4] from Chip Interrupt Controller
11	-	None	CIC3_EVT5	CIC_EVT_o [5] from Chip Interrupt Controller
12	XEVT0	McBSP0 transmit event	XEVT0	McBSP 0 Transmit Event
13	REVT0	McBSP0 receive event	REVT0	McBSP 0 Receive Event
14	XEVT1	McBSP1 transmit event	XEVT1	McBSP 1 Transmit Event
15	REVT1	McBSP1 receive event	REVT1	McBSP 1Receive Event
16	TEVTLO1	Timer 1 lower counter event	FSEVT4	Frame Synchronization Event 4
17	TEVTHI1	Timer 1 high counter event	FSEVT5	Frame Synchronization Event 5
18	-	None	FSEVT6	Frame Synchronization Event 6
19	-	None	FSEVT7	Frame Synchronization Event 7
20	INTDST1	RapidIO Interrupt 1	FSEVT8	Frame Synchronization Event 8
21	-	None	FSEVT9	Frame Synchronization Event 9
22	-	None	FSEVT10	Frame Synchronization Event 10
23	-	None	FSEVT11	Frame Synchronization Event 11
24	-	None	FSEVT12	Frame Synchronization Event 12
25	-	None	FSEVT13	Frame Synchronization Event 13
26	-	None	CIC3_EVT6	CIC_EVT_o [6] from Chip Interrupt Controller
27	-	None	CIC3_EVT7	CIC_EVT_o [7] from Chip Interrupt Controller
28	VCP2REVT	VCP2 receive event	VCPREVT	VCP Receive Event
29	VCP2XEVT	VCP2 transmit event	VCPXEVT	VCP Transmit Event
30	TCP2REVT	TCP2 receive event	TCPREVT	TCP Receive Event
31	TCP2XEVT	TCP2 transmit event	TCPXEVT	TCP Transmit Event
32	UREVT	UTOPIA receive event	SEMINT0	Semaphore Interrupt 0
33	-	None	SEMINT1	Semaphore Interrupt 1
34	-	None	SEMINT2	Semaphore Interrupt 2
35	-	None	-	Reserved
36	-	None	AIF_EVT0	AIF CPU Interrupt 0
37	-	None	AIF_EVT1	AIF CPU Interrupt 1
38	-	None	AIF_EVT2	AIF CPU Interrupt 2
39	-	None	AIF_EVT3	AIF CPU Interrupt 3
40	UXEVT	UTOPIA transmit event	AIF_PSEVT1	Packet Switched Transfer Event
41	-	None	AIF_PSEVT3	Packet Switched Transfer Event
42	-	None	AIF_PSEVT5	Packet Switched Transfer Event
43	-	None	CIC3_EVT8	CIC_EVT_o [8] from Chip Interrupt Controller
44	ICREVT	I2C receive event	IREVT	I2C Receive Event

Copyright © 2008, Texas Instruments Incorporated

Table 6. EDMA Channel Synchronization Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Event Name	Event Description	Event Name	Event Description
45	ICXEVT	I2C transmit event	IXEVT	I2C Transmit Event
46	-	None	CIC3_EVT9	CIC_EVT_o [9] from Chip Interrupt Controller
47	-	None	CIC3_EVT10	CIC_EVT_o [10] from Chip Interrupt Controller
48	GPINT0	GPIO event 0	CIC3_EVT11	CIC_EVT_o [11] from Chip Interrupt Controller
49	GPINT1	GPIO event 1	CIC3_EVT12	CIC_EVT_o [12 from Chip Interrupt Controller
50	GPINT2	GPIO event 2	CIC3_EVT13	CIC_EVT_o [13] from Chip Interrupt Controller
51	GPINT3	GPIO event 3	CIC3_EVT14	CIC_EVT_o [14] from Chip Interrupt Controller
52	GPINT4	GPIO event 4	CIC3_EVT15	CIC_EVT_o [15] from Chip Interrupt Controller
53	GPINT5	GPIO event 5	GPINT5	GPIO Event 5
54	GPINT6	GPIO event 6	GPINT6	GPIO Event 6
55	GPINT7	GPIO event 7	GPINT7	GPIO Event 7
56	GPINT8	GPIO event 8	GPINT8	GPIO Event 8
57	GPINT9	GPIO event 9	GPINT9	GPIO Event 9
58	GPINT10	GPIO event 10	GPINT10	GPIO Event 10
59	GPINT11	GPIO event 11	GPINT11	GPIO Event 11
60	GPINT12	GPIO event 12	GPINT12	GPIO Event 12
61	GPINT13	GPIO event 13	GPINT13	GPIO Event 13
62	GPINT14	GPIO event 14	GPINT14	GPIO Event 14
63	GPINT15	GPIO event 15	GPINT15	GPIO Event 15

For detailed information regarding the use of EDMA on the C6474, see the *TMS320C6474 DSP Enhanced DMA (EDMA3) Controller User's Guide* (SPRUG11).

4.7 Serial RapidIO (SRIO)

The C6455 and the C6474 both feature a SRIO interface that can be used to provide high-speed communication with other devices in the system. The SRIO interface is a high-performance, low pin-count interconnect aimed for embedded markets. Use of the SRIO interconnect in a system design can create a homogeneous interconnect environment, providing even more connectivity and control among the components. SRIO is based on the memory and device addressing concepts of processor buses where transaction processing is managed completely by hardware. This enables the SRIO interconnect to lower system cost by providing lower latency, reduced overhead of packet data processing, and higher system bandwidth, all of which are key considerations for many systems. The SRIO interconnect offers very low pin-count interfaces with scalable system bandwidth based on 10-Gigabit per second (Gbps) bidirectional links.

The physical layer (PHY) part of the SRIO peripheral consists of the physical layer and includes the input and output buffers (each serial link consists of a differential pair), the 8-bit/10-bit encoder/decoder, the PLL clock recovery, and the parallel-to-serial/serial-to-parallel converters. The SRIO interface should be designed to operate at a rate up to 3.125 Gbps per differential pair.

The SRIO interface on both devices uses the same peripheral module and the same control registers located at the same memory addresses; therefore, migration of applications using SRIO from the C6455 to the C6474 requires minimal changes in system software, however, the C6455 features four SRIO data transfer lanes, while the C6474 provides two. Besides the difference in number of data transfer lanes, the only difference in use of the SRIO interface between the C6455 and C6474 is that the memory-mapped control registers associated with the unused serial lanes on the C6474 are reserved.

Copyright © 2008, Texas Instruments Incorporated

Since the four 1x SRIO lanes on the C6455 can be configured as one 4x port as well as four 1x ports, while the two 1x lanes on the C6474 cannot, some reorganization of data transfers may be required when migrating an application using SRIO from the C6455 to the C6474.

For additional detailed information regarding the use of the C6474 SRIO, see the *TMS320C6474 Serial RapidIO (SRIO) User's Guide* (SPRUG23) and the *TMS320C6474 SERDES Implementation Guidelines* (SPRAAW9).

4.8 PCI Interface

The C6455 features a PCI interface, however the C6474 does not; therefore, for C6474 applications requiring interface with a PCI device, other provisions must be made for this in the system.

Since the PCI interface on the C6455 is a 32-bit parallel interface, and the C6474 does not provide a general-purpose external parallel interface, some type of interface adapter must be provided externally if a PCI interface is required in a C6474 system. This interface adapter must be capable of converting the PCI accesses to a type compatible with one of the available external interfaces on the C6474 such as the McBSP, EMAC or AIF. This interface adapter could be implemented externally in an field programmable gate array (FPGA) or some other form of logic, if necessary.

4.9 Multichannel Buffered Serial Port (McBSP) Interface

The C6455 and the C6474 both feature two McBSP serial port interfaces to provide connectivity to a wide variety of external devices including codecs, communications peripherals, and other processors. The same McBSP peripheral is used on both devices; therefore, migration of an application utilizing the McBSP interface from the C6455 to the C6474 requires little, if any, modification of hardware or software.

The McBSP consists of a data path and a control path that connect to external devices. Separate pins for transmission and reception communicate data to these external devices. The C64x+ CPU communicates to the McBSP using 32-bit-wide control registers accessible via the internal peripheral bus.

The primary use for the McBSP peripheral is for medium-speed external interfaces where a minimum number of device pins are required. The McBSP supports a wide variety of interface standards including AC97, IIS, IOM-2 and serial peripheral interface (SPI) modes. In addition to these interface standards, the McBSP can be programmed to support other serial formats but is not intended to be used as a high-speed interface.

The only difference between the McBSP peripherals on the C6455 and the C6474 is that on the second of the two McBSPs (McBSP1), the memory- mapped control registers are located at different addresses on the two devices. On the C6455, the McBSP1 memory-mapped registers start at address 02900000h, whereas on the C6474, the McBSP1 registers start at address 028D0000h.

For detailed information regarding the use of the McBSP on the C6474, see the *TMS320C6474 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUG17).

4.10 Phase-Locked Loop (PLL)/Clock Generators

Both the C6455 and the C6474 feature clock generators with PLLs that are used to provide clocks for these devices. While the C6455 and C6474 PLLs are slightly different in architecture, their overall features allow them to provide comparable system clocking, which allows easy migration between the two devices.

On the C6455, there are two PLLs that are used to generate a variety of different clocks for the device from two input clocks. For PLL1, the input clock can be pre-divided either by one, two or three. The resultant clock is then used as the input to the PLL, which can multiply its frequency by a programmable value from one to 32.

Either the output clock from the PLL, or, alternatively, the input clock directly, is then fed to four separate post-dividers, each of which can divide that clock by a variety of values between one and 32, generating four system clocks. Two of these output dividers use fixed divider values (three and six), and the other two are software programmable to divide by any value from one to 32. The resultant four system clocks are then used to satisfy the clocking requirements for a variety of different parts of the C6455 device.

Copyright © 2008, Texas Instruments Incorporated

Additionally, in many cases, one of these system clocks (SYSCLK4) is used to generate a device reference output clock, which is provided on the SYSCLK4/GP[1] pin, and the clock for EMIFA. Optionally, an external input clock, AECLKIN, can be used to provide the clock for EMIFA. Also, instead of SYSCLK4 being provided on the SYSCLK4/GP[1] pin, the pin can be configured to perform the General-Purpose I/O 1 (GP[1]) function instead.

Note that although many divider and multiplier factors are available for PLL1, some values may not be usable due to device minimum and maximum clock frequency requirements. Device minimum and maximum clock frequency specifications must always be met. See Section 8 of this document for additional information regarding clocking requirements for the C6455 and the C6474.

The second PLL on the C6455, PLL2, multiplies its input clock by a fixed value of 20, and the resultant clock is then divided by two. The PLL output clock divided by two is then used to provide one of the clocks required by the DDR2 memory controller.

Additionally, either the output clock from the PLL divided by two, or, alternatively, the input clock directly, is then fed to a programmable divider, which can divide that clock by a software programmable value of either two or five, to provide one of the clocks required by the EMAC.

Note that although many divider and multiplier factors are available for PLL2, some values may not be usable due to device minimum and maximum clock frequency requirements. Device minimum and maximum clock frequency specifications must always be met. See Section 8 of this document for additional information regarding clocking requirements for the C6455 and the C6474.

Figure 1 and Figure 2 show the architecture of PLL1 and PLL2 on the C6455.

Copyright © 2008, Texas Instruments Incorporated

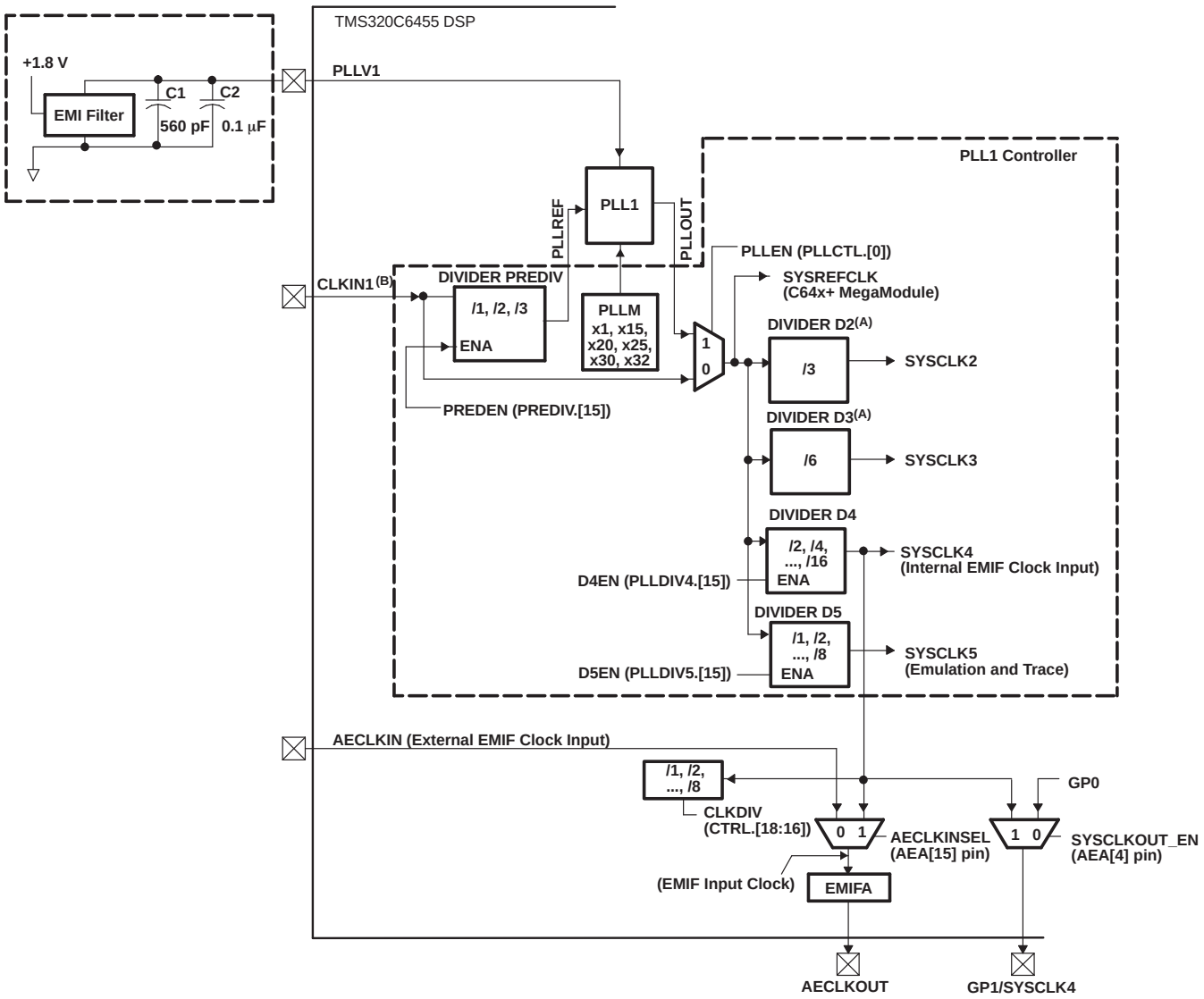

Figure 1. PLL1 and PLL1 Controller

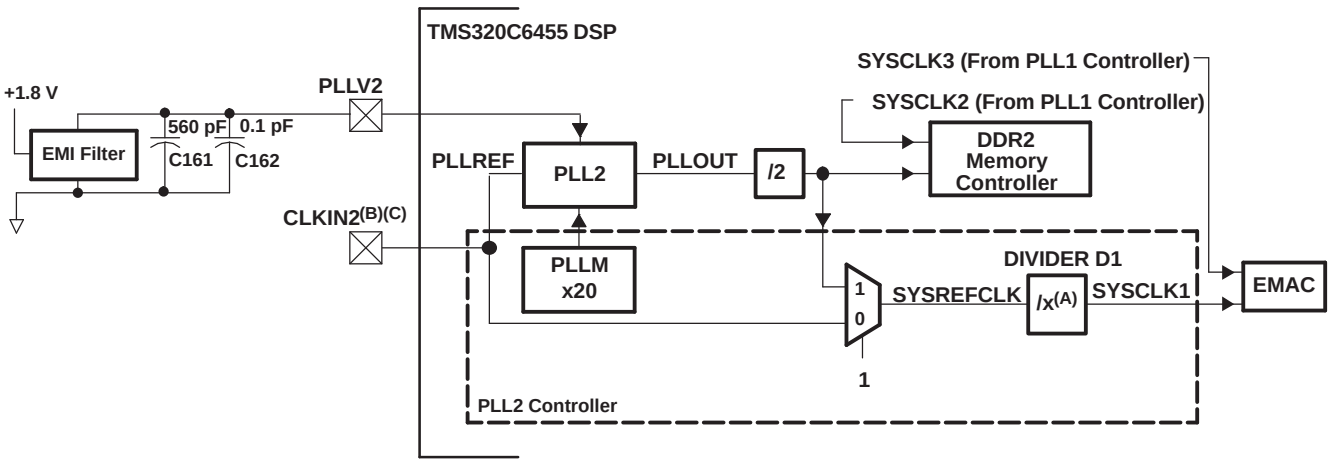

Figure 2. PLL2 Block Diagram

Copyright © 2008, Texas Instruments Incorporated

On the C6474, the clock generator also has two PLLs: PLL1 and PLL2. Each of these PLLs also supports a number of multiply and divide configurations, allowing the device to take the two input clocks and generate a number of different clock frequencies to satisfy a wide variety of system requirements.

The input clocks on the C6474 utilize differential input buffers to increase clock signal integrity and reduce sensitivity to noise. Therefore, the device input clocks must be driven by a differential clock source which is appropriately terminated. The *TMS320C6474 Hardware Design Guide* (SPRAAW7) contains additional detailed information regarding driving the clock inputs on the C6474.

For PLL1, two differential clock inputs are provided: the SYSCLKP/SYSCLKN differential input pair and the alternative clock ALTCORECLKP/ALTCORECLKN differential input pair. These two sets of inputs are provided because the SYSCLKP/N inputs are used not only as an input clock to the PLL, but also to provide clocks to the AIF SERDES, and it may be desirable to use a different input clock for these two functions. A multiplexer allows either SYSCLKP/N or ALTCORECLKP/N to be selected to be used as the input to the PLL. This multiplexer is controlled by the CORECLKSEL input pin.

Before being used to clock the PLL, the PLL1 input clock can be pre-divided by either one, two or three, programmable through software control. The resultant clock is then used as the input to the PLL, which can multiply it frequency by a software programmable value between one and 32.

The output of the PLL is then fed to eight separate post-dividers where it can be divided by various values between one and 32 to produce eight system clocks. Two of these eight post-dividers can divide by software programmable values between one and 32 or between eight and 32, while the other six dividers use preset values. The resultant output clocks from these eight post-dividers are used to provide the clocks required by most of the subsections of the C6474 device such as the C64x+ DSP cores and most of the peripherals, except the DDR2 interface.

Note that although many divider and multiplier factors are available for PLL1, some values may not be usable due to device minimum and maximum clock frequency requirements. Device minimum and maximum clock frequency specifications must always be met. See Section 8 of this document for additional information regarding clocking requirements for the C6455 and the C6474.

The second PLL on the C6474, PLL2, is used to provide clock for the DDR2 interface. The input clock for PLL2 also uses a differential input signal pair: DDRREFCLKP and DDRREFCLKN.

PLL2 multiplies its input clock by a fixed value of 10, and the resultant output is then used to provide clock to the DDR2 memory controller.

Note that migration of C6455 applications to the C6474 is further facilitated by the fact that the memory mapped PLL1 control registers are located at the same addresses on both devices, and PLL2 has no control registers on the C6474, since all aspects of its operation are fixed.

Figure 3 and Figure 4 show the architecture of PLL1 and PLL2 on the C6474.

Copyright © 2008, Texas Instruments Incorporated

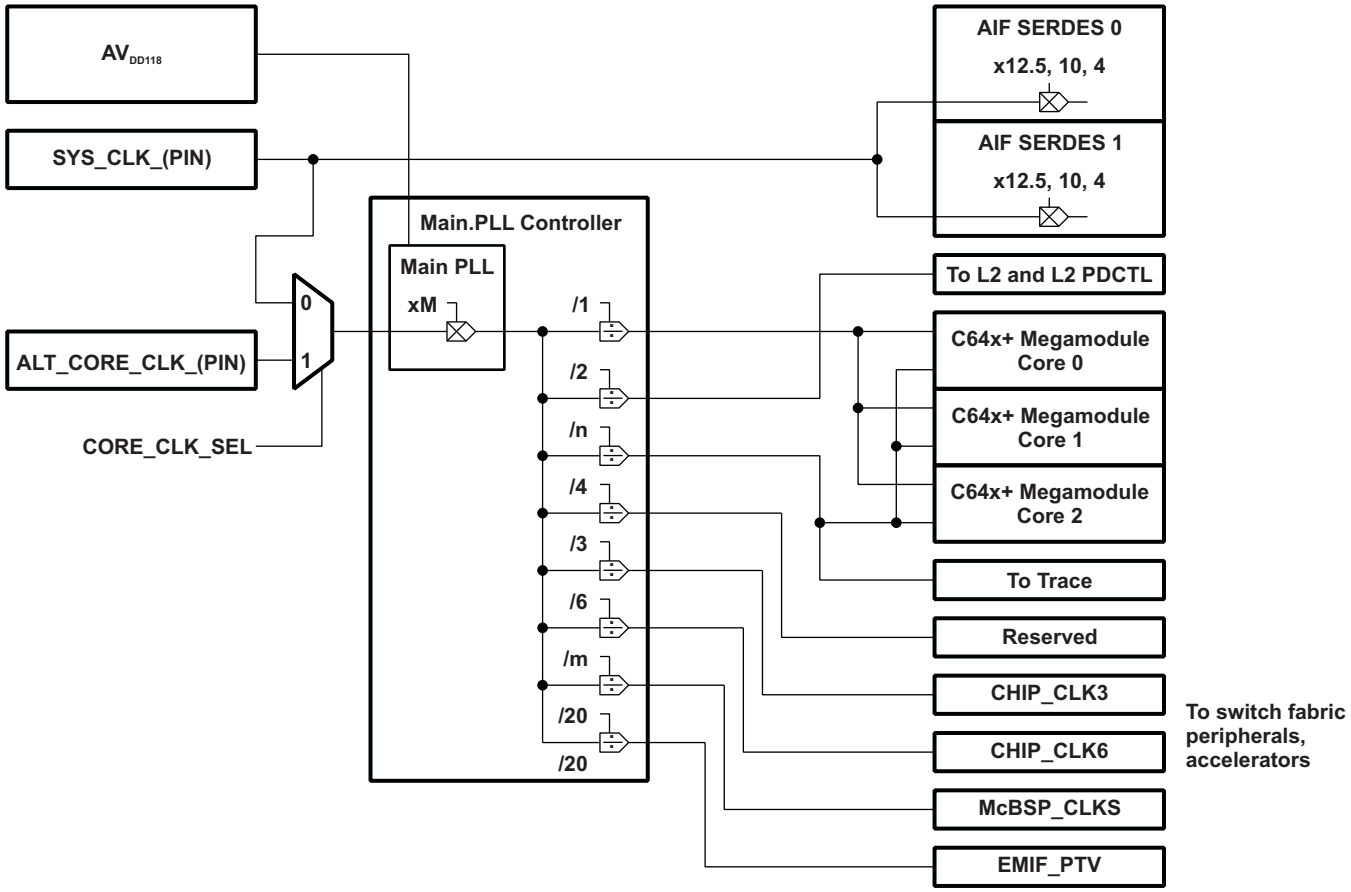

Figure 3. PLL1 and PLL1 Controller

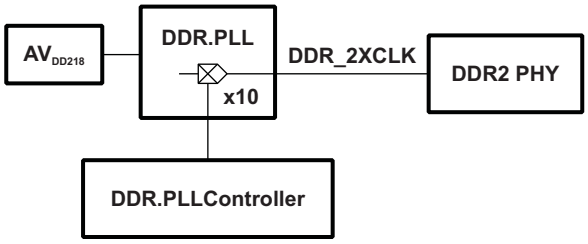

Figure 4. PLL2 Block Diagram

Copyright © 2008, Texas Instruments Incorporated

Table 7 summarizes the PLL/clock generator features on the C6455 and the C6474.

Table 7. PLL/Clock Generator Comparison

PLL	Feature	C6455	C6474
PLL1	Pre-divider ratios	1, 2, or 3	1, 2, or 3
	PLL multiplier values	1–32	1–32
	PLL output frequencies	400–1200 MHz (1)	800–1000 MHz
	Fixed output dividers	2 (3, 6)	6 (1, 2, 3, 4, 6, 20)
	Variable output dividers	2 (1–32)	2 (1–32, 8–32)
	Destinations	All but DDR2, EMAC	All but DDR2
PLL2	PLL multiplier values	Fixed: 20/2=10	Fixed: 10
	PLL output frequencies	250–533 MHz	400-667 MHz
	Output divider ratios	2 or 5	Not implemented
	Destinations	EMAC, DDR2	DDR2

(1) Range is smaller at reduced V_{DD} and in lower speed versions. See Section 10 for information on the various device speed versions power supply requirements.

Copyright © 2008, Texas Instruments Incorporated

The peripheral mix and allocation of peripheral clocking is similar on the C6455 and the C6474; however, there are several differences in clock allocation between the two devices. Table 8 presents a comparison of peripheral/module clock domain assignments for PLL1 for the C6455 and the C6474. As mentioned above, PLL2 provides clock for the EMAC and DDR2 on the C6455, and the DDR2 only on the C6474.

Table 8. Peripheral/Module Clock Domain Assignments on the C6455 and the C6474

C6455			C6474		
Ratio	Clock	Destination	Ratio	Clock	Destination
/1	SYSREFCLK	C64x+ Core	/1	SYSCLK7	C64x+ Cores
			/1	SYSCLK7	RSAs
/3	SYSCLK2	SCR	/3	SYSCLK9	SCR
/3	SYSCLK2	EDMA3	/3	SYSCLK9	EDMA3
/3	SYSCLK2	SRIO	/3	SYSCLK9	SRIO
/3	SYSCLK2	VCP2	/3	SYSCLK9	VCP2
/3	SYSCLK2	TCP2	/3	SYSCLK9	TCP2
/3	SYSCLK2	EMIFA VBUS Clocking	/3	SYSCLK9	CIC
/3	SYSCLK2	DDR2 VBUS Clocking	/3	SYSCLK9	AIF
/6	SYSCLK3	Timers	/6	SYSCLK10	Timers
/6	SYSCLK3	McBSPs	/6	SYSCLK10	McBSPs
/6	SYSCLK3	I2C	/6	SYSCLK10	I2C
/6	SYSCLK3	GPIO	/6	SYSCLK10	GPIO
/6	SYSCLK3	PLLC2	/6	SYSCLK10	PLLC2
/6	SYSCLK3	UTOPIA	/6	SYSCLK10	PLLC1
/6	SYSCLK3	PCI	/6	SYSCLK10	EMAC
/6	SYSCLK3	HPI	/6	SYSCLK10	SEMAPHORES
			/6	SYSCLK10	ROM
			/6	SYSCLK10	PSC
/n	SYSCLK5	EMU/Trace	/n	SYSCLK13	EMU/Trace
/n	SYSCLK4	EMIFA Peripheral Clocking	/n	SYSCLK11	McBSP CLKS
			/2	SYSCLK12	L2 Memory
			/20	SYSCLK14	DDR2 I/O Compensation

For information regarding initialization of the PLL/clock generators, see Section 8 of this document. For additional detailed information regarding use of the PLL/clock generator on the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C6474 DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (SPRUG09).

4.11 Timers

Both the C6455 and C6474 feature multiple 64-bit timers, each of which can also be operated as two 32-bit timers or as a watchdog timer. The same timer peripheral modules are used on both devices, however, there are some differences in using these peripherals on the two devices. On both devices, there are two timers associated with each C64x+ CPU, therefore, the C6455 has two timers, and the C6474 has six.

Since the C6455 and the C6474 both utilize the same timer peripheral modules; therefore, migration of an application utilizing the general purpose timers from the C6455 to the C6474 requires little, if any, modification of hardware or software. This section describes migration of applications from the C6455 to the C6474 device using these peripherals in detail.

Copyright © 2008, Texas Instruments Incorporated

The C6455 and C6474 general-purpose timers actually support three modes of operation: a 64-bit general-purpose (GP) timer, a dual 32-bit GP timers, or a watchdog timer. These timers can be used to generate periodic interrupts, or EDMA synchronization events. Also, the GP timers can be controlled by an external signal (either as an external input clock or to gate an internal clock), and can generate external outputs.

The predominant differences in usage between these peripheral modules on the C6455 and the C6474 are due to the fact that the C6474 has six timers, whereas the C6455 has two, and both devices have only four input/output (I/O) signals total (two inputs and two outputs) for all of the counters. Accordingly, on the C6455, when the timers are used in dual 32-bit mode, only two of the four independent timers can have inputs from external sources, while the others must use internal sources. In addition, only two of the timers can generate external outputs.

On the C6474, the two timer inputs and two timer outputs are multiplexed between the six timers (or 12 if used in dual 32-bit mode) under software control; however, there can only be two independent external inputs and two independent external outputs for all of the counters on the device. Through the multiplexing capability, any of the timers can be controlled by either of the two external inputs, and any of the 12 possible timer outputs can be directed to either of the external outputs. Externally, the C6474 can easily support any timer functionality required in migrating a C6455 application to the C6474; however, if each of the three C64x+ DSP cores on the C6474 require use of the device timers, the four external I/O signals must be shared.

Table 9 presents a comparison between the features offered by the general-purpose timers on the C6455 and the C6474.

Table 9. C6455 and C6474 General-Purpose Timer Comparison

Timer Features		C6455	C6474
Number of 64-bit timers		2	6
Support of dual 32-bit modes		Yes	Yes
Number of possible timer events	64-bit mode	2	6
	32-bit mode	4	12
Number of external clock inputs		2	2
Number of separate timer outputs		2	2

For detailed information regarding use of the C6474 general purpose timers, see the *TMS320C6474 DSP 64-Bit Timer User's Guide* (SPRUG18).

4.12 Ethernet Media Access Controller (EMAC)

The C6455 and the C6474 both feature the EMAC peripheral that provides Ethernet interface capability. Both of these devices provide support for the IEEE 802.3 compliant 10/100/1000 Mb/s Ethernet interface. The EMAC peripheral modules on these two devices have many similarities; therefore, migration of an application utilizing the EMAC interface from the C6455 to the C6474 may require only minimal modification to system software, however will require system hardware modifications. This is described in further detail later in this section.

The Ethernet interface on these devices is comprised of the EMAC and the PHY device management data input/output (MDIO) module. The EMAC controls the flow of packet data from the DSP to the PHY while the MDIO module controls PHY configuration and status monitoring.

Both the EMAC and the MDIO modules interface to the DSP through a custom interface that allows efficient data transmission and reception. This custom interface is referred to as the EMAC control module, and is considered integral to the EMAC/MDIO peripheral. The control module is also used to control device reset, interrupts, and system priority.

Copyright © 2008, Texas Instruments Incorporated

The most significant difference between the EMAC peripheral on the C6455 and the C6474 is that the C6455 supports Media Independent Interface (MII), Reduced Media Independent Interface (RMII), Gigabit Media Independent Interface (GMII) and Reduced Gigabit Media Independent Interface (RGMII) format external interfaces, while the C6474 supports only Serial Gigabit Media Independent Interface (SGMII). The SGMII external interface on the C6474 requires a different external PHY device, requiring a change in system hardware for the EMAC. At the system level, however, this provides advantages, due to its reduced number of pin connections between the C6474 and the external PHY device.

Internal to the C6474 device, software interaction with the SGMII interface is essentially the same as with the GMII interface on the C6455, simplifying software migration between the two devices. Most of the control registers' functions and addresses are the same on both devices. The only significant differences between the control registers on these two devices are that the EMAC descriptor memory is located at a different base address (02C82000h for the C6455, and 02E00000h for the C6474), and the C6455 has only two EMAC control module registers, while the C6474 has 34. The C6455 EMAC control module registers are the EMAC Control Module Interrupt Control Register (EWCTL), and the EMAC Control Module Interrupt Timer Count Register (EWINTTCNT), located at addresses 02C81004h and 02C81008, respectively. The C6474 EMAC control module registers and their memory addresses are presented in Table 10.

Table 10. EMAC Control Module Registers

Slave VBUS Address	Acronym	Register Description
0x00	IDVER	Identification and Version Register
0x04	SOFT_RESET	Soft Reset Register
0x08	EM_CONTROL	Emulation Control
0x0C	INT_CONTROL	Interrupt Control
0x10	C0_RX_THREST_EN	Core 0 Receive Threshold Interrupt Enable Register
0x14	C0_RX_EN	Core 0 Receive Interrupt Enable Register
0x18	C0_TX_EN	Core 0 Transmit Interrupt Enable Register
0x1C	C0_MISC_EN	Core 0 Misc Interrupt Enable Register
0x20	C1_RX_THRESH_EN	Core 1 Receive Threshold Interrupt Enable Register
0x24	C1_RX_EN	Core 1 Receive Interrupt Enable Register
0x28	C1_RX_EN	Core 1 Transmit Interrupt Enable Register
0x2C	C1_MISC_EN	Core 1 Misc Interrupt Enable Register
0x30	C2_RX_THRESH_EN	Core 2 Receive Threshold Interrupt Enable Register
0x34	C2_RX_EN	Core 2 Receive Interrupt Enable Register
0x38	C2_RX_EN	Core 2 Transmit Interrupt Enable Register
0x3C	C2_MISC_EN	Core 2 Misc Interrupt Enable Register
0x40	C0_RX_THRESH_STAT	Core 0 Receive Threshold Masked Interrupt Status Register
0x44	C0_RX_STAT	Core 0 Receive Interrupt Masked Interrupt Status Register
0x48	C0_TX_STAT	Core 0 Transmit Interrupt Masked Interrupt Status Register
0x4C	C0_MISC_STAT	Core 0 Misc Interrupt Masked Interrupt Status Register
0x50	C1_RX_THRESH_STAT	Core 1 Receive Threshold Masked Interrupt Status Register
0x54	C1_RX_STAT	Core 1 Receive Masked Interrupt Status Register
0x58	C1_TX_STAT	Core 1 Transmit Masked Interrupt Status Register
0x5C	C1_MISC_STAT	Core 1 Misc Masked Interrupt Status Register
0x60	C2_RX_THRESH_STAT	Core 2 Receive Threshold Masked Interrupt Status Register
0x64	C2_RX_STAT	Core 2 Receive Masked Interrupt Status Register
0x68	C2_TX_STAT	Core 2 Transmit Masked Interrupt Status Register
0x6C	C2_MISC_STAT	Core 2 Misc Masked Interrupt Status Register
0x70	C0_RX_IMAX	Core 0 Receive Interrupts Per Millisecond
0x74	C0_TX_IMAX	Core 0 Transmit Interrupts Per Millisecond
0x78	C1_RX_IMAX	Core 1 Receive Interrupts Per Millisecond
0x7C	C1_TX_IMAX	Core 1 Transmit Interrupts Per Millisecond

Copyright © 2008, Texas Instruments Incorporated

Table 10. EMAC Control Module Registers (continued)

Slave VBUS Address	Acronym	Register Description
0x80	C2_RX_IMAX	Core 2 Receive Interrupts Per Millisecond
0x84	C2_TX_IMAX	Core 2 Transmit Interrupts Per Millisecond

There are also nine unique SGMII control registers on the C6474. These registers and their addresses are presented in Table 11.

Table 11. SGMII Control Registers

Hex Address	Acronym	Register Name
02C4 0000	IDVER	Identification and Version register
02C4 0004	SOFT_RESET	Software Reset Register
02C4 0010	CONTROL	Control Register
02C4 0014	STATUS	Status Register
02C4 0018	MR_ADV_ABILITY	Advertised Ability Register
02C4 001C	-	Reserved
02C4 0020	MR_LP_ADV_ABILITY	Link Partner Advertised Ability Register
02C4 0024	-	Reserved
02C4 0030	TX_CFG	Transmit Configuration Register
02C4 0034	RX_CFG	Receive Configuration Register
02C4 0038	AUX_CFG	Auxiliary Configuration Register
02C4 0040 - 02C4 0048	-	Reserved

For detailed information regarding use of the C6474 Ethernet interface, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C6474 DSP Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) User's Guide* (SPRUG08).

4.13 Inter-Integrated Circuit (I2C) Interface

Both the C6455 and the C6474 feature an interface to I2C-compatible external devices. The I2C peripheral module provides an interface between a C64x+ DSP and other devices compliant with Philips Semiconductors Inter-IC bus (I2C bus) specification version 2.1 and connected by way of an I2C-bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the DSP through the I2C module.

The same I2C peripheral modules are used on both the C6455 and the C6474 devices; therefore, migration of an application utilizing the I2C interface from the C6455 to the C6474 requires little, if any, modification of hardware or software. This is further facilitated by the fact that all of the I2C module memory-mapped control registers are located at the same addresses in both devices.

Use of the C6474 I2C peripheral module is described in detail in the *TMS320C6474 DSP Inter-Integrated Circuit (I2C) Module User's Guide* (SPRUG22).

4.14 Host Port Interface (HPI)

The C6455 features an HPI, however, the C6474 does not; therefore, for C6474 applications requiring interface with a host device, other provisions must be made for this in the system.

Since the HPI interface on the C6455 is a 16- or 32-bit parallel interface, of which the external host is the master, and the C6474 does not provide a general-purpose external parallel interface, some type of interface adapter to the host device must be provided externally if an HPI interface is required in a C6474 system. The interface adapter must be capable of converting the host accesses to a type compatible with one of the available external interfaces on the C6474 such as the McBSP, EMAC or AIF. This interface adapter could be implemented externally in an FPGA or some other form of logic, if necessary.

Copyright © 2008, Texas Instruments Incorporated

4.15 UTOPIA Interface

The C6455 features a UTOPIA interface, however, the C6474 does not; therefore, for C6474 applications requiring compatibility with this interface type, other provisions must be made for this in the system.

Since the UTOPIA interface on the C6455 is a parallel 8-bit slave interface, and the C6474 does not provide a general-purpose external parallel interface, some type of interface adapter to the UTOPIA master device must be provided externally if UTOPIA connectivity is required in a C6474 system. The interface adapter must be capable of converting UTOPIA protocol transactions to a type compatible with one of the available external interfaces on the C6474 such as the McBSP, EMAC or AIF. This interface adapter could be implemented externally in an FPGA or some other form of logic, if necessary.

4.16 Hardware Semaphores

Since the C6474 provides three C64x+ DSP CPU cores, applications migrating from the C6455 can take advantage of this significant increase in processing power over that offered by the single core available on the C6455. To most effectively utilize the additional processing power offered by the C6474, this device includes a hardware semaphore module that can be used to arbitrate between the three DSP cores for usage of on-chip or other system resources.

The semaphore module contains 32 hardware semaphores that can be accessed by any one or all of the three DSP cores on the C6474 and allows code running on the cores to check availability of a semaphore, and request and be granted access to the semaphore if that particular semaphore is available. This function is automatically managed in hardware so that one and only one core is allowed to access a particular semaphore and become owner of the resource.

Association of a semaphore with a particular resource is not done in hardware; therefore, the semaphores can be allocated to whatever resource or resources that require arbitrated access. Because of this, whatever algorithms running on the device that require arbitrated access to a particular resource must all allocate the same semaphores for the same resources for this arbitration scheme to function properly.

Some examples of system resources that might be arbitrated using the hardware semaphores include SRIO, VCP2, TCP2, I2C, or a particular EMAC or EDMA channel.

For additional detailed information regarding function and usage of the hardware semaphores on the C6474, see the *TMS320C6474 Semaphore User's Guide* (SPRUG14).

4.17 General-Purpose Input/Output (GPIO)

Both the C6455 and the C6474 feature 16 GPIO pins that can be configured to provide independent single-bit general-purpose digital I/O. These GPIO bits can be used to interface to external signals, and to generate interrupts and EDMA synchronization events.

The C6455 and the C6474 utilize the same GPIO peripheral module; therefore, migration of an application utilizing the devices' GPIO capabilities from the C6455 to the C6474 requires little, if any, modification of hardware or software. This is further facilitated by the fact that all of the GPIO peripheral module memory-mapped control registers are located at the same addresses in both devices.

For detailed information regarding the use of GPIO on the C6474, see the *TMS320C6474 DSP General-Purpose Input/Output (GPIO) User's Guide* (SPRUG16).

5 Interrupt Considerations

The C6455 and the C6474 support servicing of a wide range of interrupts from a variety of sources, both on- and off-chip. Both devices use a multiplexing scheme to select the specific sources that will actually be allowed to interrupt their processors.

Both the C6455 and the C6474 devices support a variety of interrupts to service the needs of their many peripherals and subsystems. These interrupts are sent to the CPU as traditional interrupts, or as events called *exceptions* that are essentially the same as interrupts, but are simply interpreted more as error conditions than traditional interrupts.

Copyright © 2008, Texas Instruments Incorporated

The C64x+ DSP CPU core can accept a total of 16 possible independent interrupt inputs to the CPU, although three of these are not used, leaving a maximum of 13 interrupts actually implemented. On the CPU core, external events that are allowed to interrupt the CPU may be chosen from a possible selection of 128 system events, in addition to three events that are always selected, allocated to one hardware exception (decoded from the 128 system events), reset (that is neither an interrupt nor an exception), and NMI, which can be an interrupt or an exception. Some of the possible interrupts can be generated from GPIO signals, and the polarities of these signals used to generate the interrupts are programmable through the GPIO control registers.

The 128 possible system events are mapped to 12 independent interrupts and one hardware exception using the interrupt selector, interrupt combiner, and exception combiner modules. The following resultant 15 interrupt and exception signals are sent to the CPU:

- Twelve selected/combined interrupts
- Reset (neither interrupt nor exception)
- NMI (can be either interrupt or exception)
- One Hardware Exception

The 12 interrupt signals along with NMI form the 13 signals allowed to generate interrupts to the CPU core. The status of these signals is reflected in 13 bits in the Interrupt Flag Register (IFR), indicating which of these interrupts is active.

The NMI signal and the hardware exception signal are sent to the Exception Flag Register (EFR), that reflects which of these events is active, along with two possible internal exceptions: the software exception and the internal exception. The status of the reset input signal is not reflected in either the IFR or EFR.

Figure 5 shows a functional representation of the C64x+ core interrupt controller interface.

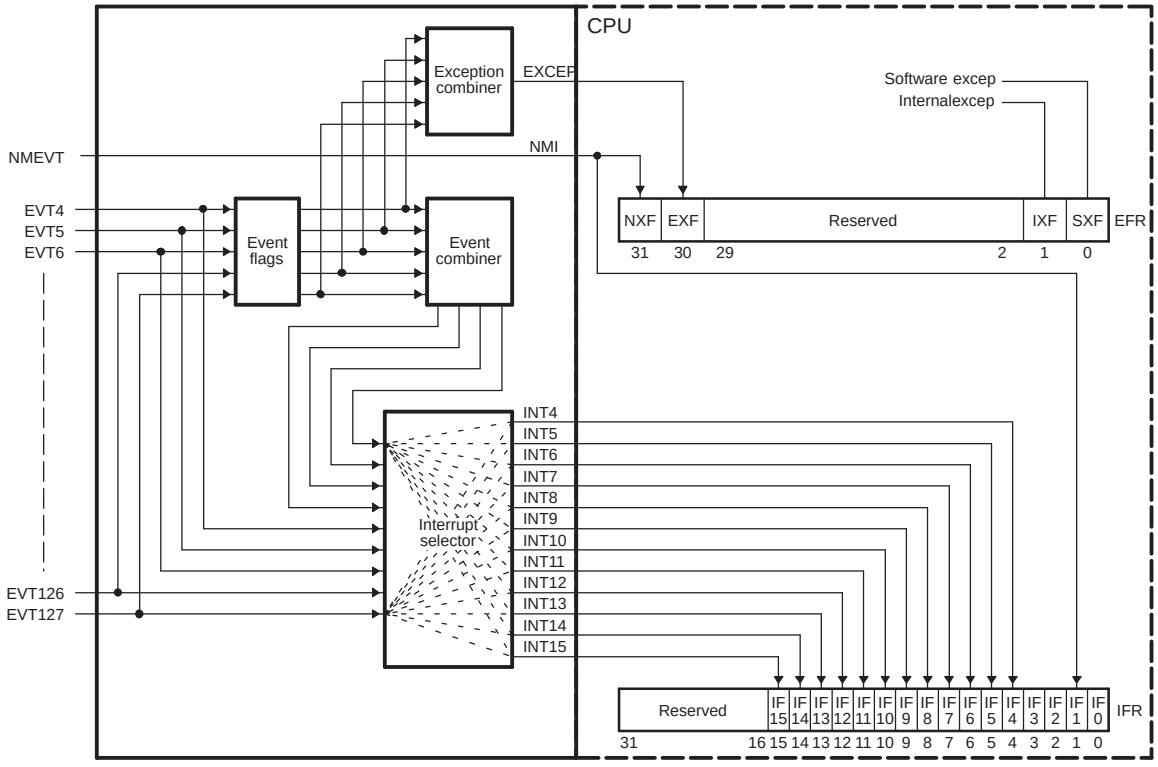

Figure 5. C64x+ Core Interrupt Controller Interface

Copyright © 2008, Texas Instruments Incorporated

On the C6474, since there are three C64x+ DSP CPU cores, as opposed to one on the C6455, and additional peripherals, the interrupt structure on this device has some additional features beyond those present on the C6455. Each DSP core on the C6474 contains a separate interrupt controller that functions in the same fashion as the interrupt controller on the C6455, however, since there are additional peripherals on the C6474, and additional interrupt sources, additional interrupt control and multiplexing was added to this device.

The additional interrupt control and multiplexing on the C6474 is performed by a functional unit called the chip interrupt controller (CIC). There are four on the device: one associated with each C64x+ core and one associated with EDMA TPCC. Each of the CICs further multiplexes 16 of the 128 possible interrupt sources on the device to each of the cores (or the EDMA TPCC) based on a structure that is essentially the same as the interrupt controllers internal to each core. Each CIC has interrupt inputs from up to 64 sources, and can select 16 of these sources to provide interrupts to its associated core. The 64 possible sources are mapped to 16 independent interrupts using an event selector, and an event combiner, which can combine any of the sources into two interrupt inputs. The resultant interrupt signals are sent to the appropriate CPU core interrupt controller as 16 of the 128 possible interrupt sources.

Table 12 presents a comparison of the interrupt capabilities of the C6455 and the C6474 devices.

Table 12. Interrupt Capability Comparison

Features	C6455	C6474 (1)
Total # of sources	128	128 + 64
Total # CPU interrupts	16	16 (per CPU)
Fixed	3: Reset, NMI, H/W exception	3: Reset, NMI, hardware exception (1)
Programmable	12	12 (per CPU)
Source selected by	Interrupt/exception selector/combiners	Interrupt/exception selector/combiners
Selectable polarity	GPIO interrupts only	GPIO interrupts only
CICs		4 (one per CPU + TPCC)
CIC Interrupt Inputs		64
CIC Interrupt Outputs		16 (to each CPU and TPCC)

(1) Note that each core on the C6474 has its own separate NMI input and hardware exception.

The 128 system event sources, from which the 12 interrupts routed to the CPU are chosen, are hard-wired on both the C6455 and the C6474. The interrupts that are actually routed to the CPU in each device are chosen from this selection as described above. This interrupt selection structure allows for maximum flexibility within the system for allocation of necessary interrupt servicing.

Table 13 presents a comparison of the 128 possible system events for generating interrupts available on the C6455 and the C6474. As described above, the C6474 has 64 additional interrupts that can also be multiplexed to 16 interrupt inputs on each of the C64x+ cores and the EDMA TPCC using the four CICs available on the device. The first three CICs, CIC[2:0], multiplex interrupts for each of the three DSP CPU cores on the C6474, while CIC[3] multiplexes interrupts for the EDMA3 TPCC.

Table 13. Interrupt System Event Comparison

EDMA Channel	C6455		C6474	
	Interrupt Event	Event Description	Interrupt Event	Event Description
0	EVT0	Output of event combiner 0 in interrupt controller, for events 1 -31.	EVT0	Output of Event Combiner 0 for Events [31:4]
1	EVT1	Output of event combiner 1 in interrupt controller, for events 32 -63.	EVT1	Output of Event Combiner 1 for Events [63:32]
2	EVT2	Output of event combiner 2 in interrupt controller, for events 64 -95.	EVT2	Output of Event Combiner 2 for Events [95:64]
3	EVT3	Output of event combiner 3 in interrupt controller, for events 96 -127.	EVT3	Output of Event Combiner 3 for Events [127:96]
4	Reserved	Reserved. These system events are not connected and, therefore, not used.	SEMINTn	Semaphore Grant Interrupt
5	Reserved	Reserved. These system events are not connected and, therefore, not used.	MACINTn	Ethernet MAC Control Interrupt

Copyright © 2008, Texas Instruments Incorporated

Table 13. Interrupt System Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Interrupt Event	Event Description	Interrupt Event	Event Description
6	Reserved	Reserved. These system events are not connected and, therefore, not used.	MACRXINTn	Ethernet MAC Receive Interrupt
7	Reserved	Reserved. These system events are not connected and, therefore, not used.	MACTXINTn	Ethernet MAC Transmit Interrupt
8	Reserved	Reserved. These system events are not connected and, therefore, not used.	MACTHRESHn	Ethernet MAC Receive Threshold Interrupt
9	EMU_DTDMA	EMU interrupt for: 1. Host scan access 2. DTDMA transfer complete 3. AET interrupt	EMU_DTDMA	ECM Interrupt for: 1. Host Scan Access 2. DTDMA Transfer Complete 3. AET Interrupt
10	None	This system event is not connected and, therefore, not used.	RAC INTn	RAC Interrupt N
11	EMU_RTDXRX	EMU real-time data exchange (RTDX) receive complete	EMU_RTDXRX	RTDX Receive Complete
12	EMU_RTDXTX	EMU RTDX transmit complete	EMU_RTDXTX	RTDX Transmit Complete
13	IDMA0	IDMA channel 0 interrupt	IDMAINT0	IDMA Channel 0 Interrupt
14	IDMA1	IDMA channel 1 interrupt	IDMAINT1	IDMA Channel 1 Interrupt
15	DSPINT	HPI/PCI-to-DSP interrupt	FSEVT0	Frame Synchronization Event 0
16	I2CINT	I2C interrupt	FSEVT1	Frame Synchronization Event 1
17	MACINT	Ethernet MAC interrupt	FSEVT2	Frame Synchronization Event 2
18	AEASYNCERR	EMIFA error interrupt	FSEVT3	Frame Synchronization Event 3
19	Reserved	Reserved. This system event is not connected and, therefore, not used.	FSEVT4	Frame Synchronization Event 4
20	INTDST0	RapidIO interrupt 0	FSEVT5	Frame Synchronization Event 5
21	INTDST1	RapidIO interrupt 1	FSEVT6	Frame Synchronization Event 6
22	INTDST4	RapidIO interrupt 4	FSEVT7	Frame Synchronization Event 7
23	Reserved	Reserved. This system event is not connected and, therefore, not used.	FSEVT8	Frame Synchronization Event 8
24	EDMA3CC_GINT	EDMA3 channel global completion interrupt	FSEVT9	Frame Synchronization Event 9
25	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT10	Frame Synchronization Event 10
26	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT11	Frame Synchronization Event 11
27	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT12	Frame Synchronization Event 12
28	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT13	Frame Synchronization Event 13
29	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT14	Frame Synchronization Event 14
30	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT15	Frame Synchronization Event 15
31	Reserved	Reserved. These system events are not connected and, therefore, not used.	FSEVT16	Frame Synchronization Event 16
32	VCP2_INT	VCP2 error interrupt	FSEVT17	Frame Synchronization Event 17
33	TCP2_INT	TCP2 error interrupt	TINT0L	Timer 0 Interrupt Low
34	Reserved	Reserved. These system events are not connected and, therefore, not used.	TINT0H	Timer 0 Interrupt High
35	Reserved	Reserved. These system events are not connected and, therefore, not used.	TINT1L	Timer 1 Interrupt Low
36	UINT	McBSP0 transmit interrupt	TINT1H	Timer 1 Interrupt High
37	Reserved	McBSP1 receive interrupt	TINT2L	Timer 2 Interrupt Low
38	Reserved	McBSP1 transmit interrupt	TINT2H	Timer 2 Interrupt High
39	Reserved	Reserved. These system events are not connected and, therefore, not used.	TINT3L	Timer 3 Interrupt Low
40	RINT0	McBSP0 receive interrupt	TINT3H	Timer 3 Interrupt High
41	XINT0	McBSP0 transmit interrupt	TINT4L	Timer 4 Interrupt Low

Copyright © 2008, Texas Instruments Incorporated

Table 13. Interrupt System Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Interrupt Event	Event Description	Interrupt Event	Event Description
42	RINT1	McBSP1 receive interrupt	TINT4H	Timer 4 Interrupt High
43	XINT1	McBSP1 transmit interrupt	TINT5L	Timer 5 Interrupt Low
44	Reserved	Reserved. Do not use.	TINT5H	Timer 5 Interrupt High
45	Reserved	Reserved. Do not use.	GPINT0	GPIO Interrupt 0
46	Reserved	Reserved. Do not use.	GPINT1	GPIO Interrupt 1
47	Reserved	Reserved. Do not use.	GPINT2	GPIO Interrupt 2
48	Reserved	Reserved. Do not use.	GPINT3	GPIO Interrupt 3
49	Reserved	Reserved. Do not use.	GPINT4	GPIO Interrupt 4
50	Reserved	Reserved. Do not use.	GPINT5	GPIO Interrupt 5
51	GPINT0	GPIO interrupt	GPINT6	GPIO Interrupt 6
52	GPINT1	GPIO interrupt	GPINT7	GPIO Interrupt 7
53	GPINT2	GPIO interrupt	GPINT8	GPIO Interrupt 8
54	GPINT3	GPIO interrupt	GPINT9	GPIO Interrupt 9
55	GPINT4	GPIO interrupt	GPINT10	GPIO Interrupt 10
56	GPINT5	GPIO interrupt	GPINT11	GPIO Interrupt 11
57	GPINT6	GPIO interrupt	GPINT12	GPIO Interrupt 12
58	GPINT7	GPIO interrupt	GPINT13	GPIO Interrupt 13
59	GPINT8	GPIO interrupt	GPINT14	GPIO Interrupt 14
60	GPINT9	GPIO interrupt	GPINT15	GPIO Interrupt 15
61	GPINT10	GPIO interrupt	TPCC_GINT	EDMA Channel Global Completion Interrupt
62	GPINT11	GPIO interrupt	TPCC_INT0	TPCC Completion Interrupt -Mask 0
63	GPINT12	GPIO interrupt	TPCC_INT1	TPCC Completion Interrupt -Mask 1
64	GPINT13	GPIO interrupt	TPCC_INT2	TPCC Completion Interrupt -Mask 2
65	GPINT14	GPIO interrupt	TPCC_INT3	TPCC Completion Interrupt -Mask 3
66	GPINT15	GPIO interrupt	TPCC_INT4	TPCC Completion Interrupt -Mask 4
67	TINTLO0	Timer 0 lower counter interrupt	TPCC_INT5	TPCC Completion Interrupt -Mask 5
68	TINTHI0	Timer 0 higher counter interrupt	TPCC_INT6	TPCC Completion Interrupt -Mask 6
69	TINTLO1	Timer 1 lower counter interrupt	TPCC_INT7	TPCC Completion Interrupt -Mask 7
70	TINTHI1	Timer 1 higher counter interrupt	Unused	Reserved
71	EDMA3CC_INT0	EDMA3CC completion interrupt -Mask0	RIOINT (2n)(4)	RapidIO Interrupt (2n)
72	EDMA3CC_INT1	EDMA3CC completion interrupt -Mask1	RIOINT (2n+1)(4)	RapidIO Interrupt (2n+1)
73	EDMA3CC_INT2	EDMA3CC completion interrupt -Mask2	AIF_EVT0	Error/Alarm Event 0
74	EDMA3CC_INT3	EDMA3CC completion interrupt -Mask3	AIF_EVT1	Error/Alarm Event 1
75	EDMA3CC_INT4	EDMA3CC completion interrupt -Mask4	Unused	Reserved
76	EDMA3CC_INT5	EDMA3CC completion interrupt -Mask5	IPC_LOCAL	Inter DSP Interrupt from IPCGRn
77	EDMA3CC_INT6	EDMA3CC completion interrupt -Mask6	Unused	Reserved
78	EDMA3CC_INT7	EDMA3CC completion interrupt -Mask7	Unused	Reserved
79	EDMA3CC_ERRINT	EDMA3CC error interrupt	Unused	Reserved
80	Reserved	Reserved. This system event is not connected and, therefore, not used.	CICn_EVT0	System Event 0 (Combined) from Chip Interrupt Controller[n](5)
81	EDMA3TC0_ERRINT	EDMA3TC0 error interrupt	CICn_EVT1	System Event 1 (Combined) from Chip Interrupt Controller[n]
82	EDMA3TC1_ERRINT	EDMA3TC1 error interrupt	CICn_EVT2	System Event 2 from Chip Interrupt Controller[n]
83	EDMA3TC2_ERRINT	EDMA3TC2 error interrupt	CICn_EVT3	System Event 3 from Chip Interrupt Controller[n]
84	EDMA3TC3_ERRINT	EDMA3TC3 error interrupt	CICn_EVT4	System Event 4 from Chip Interrupt Controller[n]
85	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT5	System Event 5 from Chip Interrupt Controller[n]
86	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT6	System Event 6 from Chip Interrupt Controller[n]

Copyright © 2008, Texas Instruments Incorporated

Table 13. Interrupt System Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Interrupt Event	Event Description	Interrupt Event	Event Description
87	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT7	System Event 7 from Chip Interrupt Controller[n]
88	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT8	System Event 8 from Chip Interrupt Controller[n]
89	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT9	System Event 9 from Chip Interrupt Controller[n]
90	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT10	System Event 10 from Chip Interrupt Controller[n]
91	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT11	System Event 11 from Chip Interrupt Controller[n]
92	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT12	System Event 12 from Chip Interrupt Controller[n]
93	Reserved	Reserved. These system events are not connected and, therefore, not used.	CICn_EVT13	System Event 13 from Chip Interrupt Controller[n]
94	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
95	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
96	INTERR	Interrupt Controller dropped CPU interrupt event	INTERR	Dropped CPU Interrupt Event
97	EMC_IDMAERR	EMC invalid IDMA parameters	EMC_IDMAERR	Invalid IDMA Parameters
98	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
99	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
100	EFIINTA	EFI interrupt from side A	EFINTA	EFI Interrupt from Side A
101	EFIINTB	EFI interrupt from side B	EFIINTB	EFI Interrupt from Side B
102	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
103	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
104	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
105	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
106	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
107	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
108	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
109	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
110	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
111	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
112	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
113	L1P_ED1	L1P single bit error detected during DMA read	PMC_ED	Single Bit Error Detected during DMA Read
114	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
115	Reserved	Reserved. These system events are not connected and, therefore, not used.	Unused	Reserved
116	L2_ED1	L2 single bit error detected	UMC_ED1	Corrected Bit Error Detected
117	L2_ED2	L2 two bit error detected	UMC_ED2	Uncorrected Bit Error Detected
118	PDC_INT	Powerdown sleep interrupt	PDC_INT	PDC Sleep Interrupt
119	Reserved	Reserved. These system events are not connected and, therefore, not used.	SYS_CMPA	CPU Memory Protection Fault

Copyright © 2008, Texas Instruments Incorporated

Table 13. Interrupt System Event Comparison (continued)

EDMA Channel	C6455		C6474	
	Interrupt Event	Event Description	Interrupt Event	Event Description
120	L1P_CMPA	L1P CPU memory protection fault	PMC_CMPA	CPU Memory Protection Fault
121	L1P_DMPA	L1P DMA memory protection fault	PMC_DMPA	DMA Memory Protection Fault
122	L1D_CMPA	L1D CPU memory protection fault	DMC_CMPA	CPU Memory Protection Fault
123	L1D_DMPA	L1D DMA memory protection fault	DMC_DMPA	DMA Memory Protection Fault
124	L2_CMPA	L2 CPU memory protection fault	UMC_CMPA	CPU Memory Protection Fault
125	L2_DMPA	L2 DMA memory protection fault	UMC_DMPA	DMA Memory Protection Fault
126	IDMA_CMPA	IDMA CPU memory protection fault	EMC_CMPA	CPU Memory Protection Fault
127	IDMA_BUSERR	IDMA bus error interrupt	EMC_BUSERR	Bus Error Interrupt

Table 14 presents a list of the 64 additional possible interrupt inputs further multiplexed by CIC[2:0] on the C6474.

Table 14. C6474 Chip Interrupt Controller CIC[2:0] System Event Allocations

Event Channel	Event	Event Description
0	EVT0	Output of Event Controller 0 for Events [31:2]
1	EVT1	Output of Event Controller 1 for Events [63:32]
2	Unused	Reserved
3	Unused	Reserved
4	I2CINT	Error Interrupt
5	FSERR1	Error/Alarm Interrupt 1
6	RIOINT7	Error Interrupt
7	FSERR2	Error/Alarm Interrupt 2
8	VCPINT	Error Interrupt
9	TCPINT	Error Interrupt
10	RINT0	McBSP0 Receive Interrupt
11	XINT0	McBSP0 Transmit Interrupt
12	RINT1	McBSP1 Receive Interrupt
13	XINT1	McBSP1 Transmit Interrupt
14	REVT0	McBSP0 Receive EDMA Event
15	XEVT0	McBSP0 Transmit EDMA Event
16	REVT1	McBSP1 Receive EDMA Event
17	XEVT1	McBSP1 Transmit EDMA Event
18	IREVT	I2C Receive EDMA Event
19	IXEVT	I2C Transmit EDMA Event
20	FSEVT18	FSYNC Event 18
21	FSEVT19	FSYNC Event 19
22	FSEVT20	FSYNC Event 20
23	FSEVT21	FSYNC Event 21
24	FSEVT22	FSYNC Event 22
25	FSEVT23	FSYNC Event 23
26	FSEVT24	FSYNC Event 24
27	FSEVT25	FSYNC Event 25
28	FSEVT26	FSYNC Event 26
29	FSEVT27	FSYNC Event 27
30	FSEVT28	FSYNC Event 28
31	FSEVT29	FSYNC Event 28
32	VCPREVT	VCP Receive Event

Copyright © 2008, Texas Instruments Incorporated

Table 14. C6474 Chip Interrupt Controller CIC[2:0] System Event Allocations (continued)

Event Channel	Event	Event Description
33	VCPXEVT	VCP Transmit Event
34	TCPREVT	TCP Receive Event
35	TCPXEVT	TCP Transmit Event
36	TPCC_ERRINT	TPCC Error Interrupt
37	TPCC_MPINT	TPCC Memory Protection Interrupt
38	TPTC_ERRINT0	TPTC0 Error Interrupt
39	TPTC_ERRINT1	TPTC1 Error Interrupt
40	TPTC_ERRINT2	TPTC2 Error Interrupt
41	TPTC_ERRINT3	TPTC3 Error Interrupt
42	TPTC_ERRINT4	TPTC4 Error Interrupt
43	TPTC_ERRINT5	TPTC5 Error Interrupt
44	TPCC_AETEVT	TPCC AET Event
45	AIF_EVT2	AIF CPU Interrupt 2
46	AIF_EVT3	AIF CPU Interrupt 2
47	AIF_PSEVT0	Packet Switched Transfer Event 0
48	AIF_PSEVT1	Packet Switched Transfer Event 1
49	AIF_PSEVT2	Packet Switched Transfer Event 2
50	AIF_PSEVT3	Packet Switched Transfer Event 3
51	AIF_PSEVT4	Packet Switched Transfer Event 4
52	AIF_PSEVT5	Packet Switched Transfer Event 5
53	AIF_PSEVT6	Packet Switched Transfer Event 6
54	AIF_BUFEVT	AIF Capture Buffer Event.
55	Unused	Reserved
56	RAC_DEVENT0	Debug Event
57	RAC_DEVENT1	Debug Event
58	SEMERR*n* (1)	Semaphore Error Event for C64x+ Megamodule*n*
59 - 63	Unused	Reserved

(1) C64x+ Megamodule Core 0, C64x+ Megamodule Core 1, and C64x+ Megamodule Core 2 receive SEMERR0, SEMERR1, and SEMERR2, respectively.

Table 15 presents a list of the 64 additional possible interrupt inputs further multiplexed by CIC[3] on the C6474.

Table 15. TPCC Interrupt Controller Event List CIC[3]

Event Channel	Event	Event Description
0	EVT0	Output of Event Controller 0 for Events [31:2]
1	EVT1	Output of Event Controller 1 for Events [63:32]
2	FSEVT0	Frame Synchronization Event 0
3	FSEVT1	Frame Synchronization Event 1
4	FSEVT2	Frame Synchronization Event 2
5	FSEVT3	Frame Synchronization Event 3
6	FSEVT14	Frame Synchronization Event 14
7	FSEVT15	Frame Synchronization Event 15
8	FSEVT16	Frame Synchronization Event 16
9	FSEVT17	Frame Synchronization Event 17
10	FSEVT18	Frame Synchronization Event 18
11	FSEVT19	Frame Synchronization Event 19
12	FSEVT20	Frame Synchronization Event 20

Copyright © 2008, Texas Instruments Incorporated

Table 15. TPCC Interrupt Controller Event List CIC[3] (continued)

Event Channel	Event	Event Description
13	FSEVT21	Frame Synchronization Event 21
14	FSEVT22	Frame Synchronization Event 22
15	FSEVT23	Frame Synchronization Event 23
16	FSEVT24	Frame Synchronization Event 24
17	FSEVT25	Frame Synchronization Event 25
18	FSEVT26	Frame Synchronization Event 26
19	FSEVT27	Frame Synchronization Event 27
20	FSEVT28	Frame Synchronization Event 28
21	RIOINT0	RapidIO Interrupt 0
22	RIOINT1	RapidIO Interrupt 1
23	RIOINT2	RapidIO Interrupt 2
24	RIOINT3	RapidIO Interrupt 3
25	RIOINT4	RapidIO Interrupt 4
26	RIOINT5	RapidIO Interrupt 5
27	RIOINT7	RapidIO Interrupt 7
28	MACINT0	Ethernet EMAC Interrupt
29	MACRINT0	Ethernet EMAC Interrupt
30	MACXINT0	Ethernet EMAC Interrupt
31	MACINT1	Ethernet EMAC Interrupt
32	MACRINT1	Ethernet EMAC Interrupt
33	MACXINT1	Ethernet EMAC Interrupt
34	MACINT2	Ethernet EMAC Interrupt
35	MACRINT2	Ethernet EMAC Interrupt
36	MACXINT2	Ethernet EMAC Interrupt
37	SEMERR0	Semaphore Error Interrupt
38	SEMERR1	Semaphore Error Interrupt
39	SEMERR2	Semaphore Error Interrupt
40 - 42	Unused	Reserved
43	TINT3L	Timer Interrupt Low
44	TINT3H	Timer Interrupt High
45	TINT4L	Timer Interrupt Low
46	TINT4H	Timer Interrupt High
47	TINT5L	Timer Interrupt Low
48	TINT5H	Timer Interrupt High
49	AIF_BUFEVT	AIF Capture Buffer Event
50	FSEVT29	Frame Synchronization Event 29
51 - 52	Unused	Reserved
53	GPINT0	GPIO Event
54	GPINT1	GPIO Event
55	GPINT2	GPIO Event
56	GPINT3	GPIO Event
57	GPINT4	GPIO Event
58	CIC0_EVT14	CIC_EVT_o[14] from Chip Interrupt Controller[0]
59	CIC0_EVT15	CIC_EVT_o[15] from Chip Interrupt Controller[0]
60	CIC1_EVT14	CIC_EVT_o[14] from Chip Interrupt Controller[1]
61	CIC1_EVT15	CIC_EVT_o[15] from Chip Interrupt Controller[1]
62	CIC2_EVT14	CIC_EVT_o[14] from Chip Interrupt Controller[2]

Copyright © 2008, Texas Instruments Incorporated

Table 15. TPCC Interrupt Controller Event List CIC[3] (continued)

Event Channel	Event	Event Description
63	CIC2_EVT15	CIC_EVT_o[15] from Chip Interrupt Controller[2]

For detailed information regarding handling of interrupts on the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552), the *TMS320C64x+ DSP Megamodule Reference Guide* (SPRU871), the *TMS320C6474 DSP Chip Interrupt Controller (CIC) User's Guide* (SPRUFK6), and the *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (SPRU732).

6 Bootloading Capabilities

The C6455 and the C6474 provide the capability to transfer code from an external location into RAM to be executed following reset. On both devices, the states of various input pins are sampled during reset, and the selected boot modes are determined based on these states. Both devices offer a similar group of bootload modes; however, the C6474 offers additional functionality including secure bootloading capability with each of the basic bootloader modes.

On the C6455, there are six possible boot modes, which are primarily controlled by the BOOTMODE[3:0] input pins. These six boot modes are comprised of an 8-bit EMIF boot, a host boot from either HPI or PCI, I2C boot in either master or slave mode, SRIO boot, and a *no boot* mode that simply branches to address 0800000h in L2 SRAM.

In EMIF-boot mode, the bootloader assumes that the code to be executed is already located in 8-bit external memory, and begins execution at location B0000000h in CE3 address space.

In HPI or PCI host-boot mode, the CPU waits for the external host to load the code to be executed into memory. Once the host has finished loading the code, it must generate an interrupt, which causes the CPU to begin execution at address 0800000h in L2 SRAM. For HPI host-boot mode, the PCI_EN input pin must be low. For PCI host-boot mode, the PCI_EN input pin must be high, and the CFGGP[2:0] pins must be zero.

In the master I2C-boot mode, the C6455 functions as an I2C master device, and loads code from an external I2C slave device into memory to be executed. Information about where the code is to be located, and how the C6455 is to be configured during and following the boot operation is contained in the initial data loaded from the boot module. The bootloader requires the I2C slave address to be 50h.

In slave I2C-boot mode, the C6455 functions as an I2C slave device, and waits for an external I2C master device to download the code to be executed into the DSP. As with the master I2C-boot mode, information about where the code is to be located, and how the C6455 is to be configured during and following the boot operation is contained in the initial data loaded from the boot module.

The C6455 can also bootload code from the SRIO interface using a variety of different configurations. In addition to the BOOTMODE[3:0] pins, the CFGGP[2:0] input pins are also used in SRIO-boot mode to select the desired device ID. The device ID is a four-bit field consisting of BOOTMODE2 as the MSB, and CFGGP[2:0] as the LSBs, allowing for 16 unique device IDs.

On the C6455, boot modes are classified as either hardware- or software-boot modes depending on whether code from the ROM is actually executed to implement the selected boot function, or whether the function is implemented through the boot configuration logic hardware. Only the *no boot* and EMIF-boot modes are hardware-boot modes on the C6455; all of the other boot modes are software-boot modes. For all of the software-boot modes, the CPU clock frequency must not exceed 750 MHz.

Also, on the C6455, all of the boot modes except the EMIF- and SRIO-boot modes have the main PLL (PLL1) configured in bypass mode. In the EMIF- and SRIO-boot modes, PLL1 is configured in multiply-by-15 mode.

The C6474 provides 11 basic bootloader modes, which are each available in either secure or public mode. The 11 bootloader modes on the C6474 are essentially the same modes available on the C6455 with the same basic features, but with a few additions and a few operational differences. Public mode involves bootloading without code being made secure in any fashion. In secure mode, the bootloader transfers code in each of the bootload modes in the same fashion functions as in public mode, with the added capability to support information security. Secure bootloading mode is discussed in further detail later in this section.

Copyright © 2008, Texas Instruments Incorporated

For the slave I2C-boot mode, bootloader operation is essentially the same as on the C6455. For the master I2C-boot mode, operation is essentially the same as on the C6455, however, an additional option is provided on the C6474 allowing this device the capability to also boot from a device with a slave address of 51h. SRIO-boot and *no boot*-modes are also supported on the C6474.

The C6474 does not implement the HPI, PCI, or EMIF interfaces, therefore, these boot modes are not supported on this device.

In addition to the same basic boot modes supported on the C6455, the C6474 also offers the additional capability of being able to boot through the EMAC in three different modes: master mode, slave mode, and forced-link mode.

All of 11 boot modes on the C6474 are selected using unique codes on the BOOTMODE[3:0] inputs, which are read from the GPIO[3:0] input pins at boot time. The CFG[3:0] inputs, which are read from the GPIO[11:8] input pins at boot time, are used to provide additional detailed configuration information in several of the boot modes. For all boot modes on the C6474, the main PLL (PLL1) is initialized to a multiply-by-16 configuration.

For secure bootloading, as mentioned earlier, capability is provided to support information security. Two mechanisms are provided to support information security: on-chip memory security and encryption/decryption of bootloaded information/code. Both of these mechanisms are implemented through the use of EFUSE programming performed at the time of production test for each device, and are available only by special arrangement with Texas Instruments. To arrange for either or both of these options, contact your local Texas Instruments representative or distributor.

On-chip memory security is a mechanism through which the contents of internal memory on the device is not visible from outside the device. On-chip memory security is enabled by programming an EFUSE at the time of production test for each device, if selected. Once this EFUSE is programmed, the device is configured as a secure device, and internal memory cannot be accessed by any external device, including with an emulator.

Note that on a secure device, information is not protected if transmitted to an external device through one of the peripheral interfaces. Also note that information stored in external DDR2 memory is not protected, and can be observed as this memory is read or written to.

For encrypted bootloading, information (code and/or data) to be bootloaded must first be encrypted before being bootloaded. This encryption is performed using the standard DES encryption algorithm using a key agreed upon with the customer corresponding to a decryption key programmed into the device using EFUSE capability. The EFUSEs for the encryption/decryption algorithm key are also programmed at the time of final test for each device, in the same fashion as for a secure memory device. Once these EFUSEs are programmed, the bootloader uses this key to decrypt the bootloaded information before loading into memory, so that code and data are returned to their un-encrypted form for use.

During secure bootloading, all three C64x+ DSP cores are released from reset and begin executing from secure ROM. Software in the secure ROM allocates internal RAM pages, after which DSP core 0 initiates the boot process and the other two cores will wait. DSP core 0 performs any authentication and decryption required on the bootloaded image prior to releasing the other cores to begin execution. After the secure loading is complete, DSP core 0 releases the other two cores. Then, DSP core 0 begins execution from the entry address defined in the boot table. DSP cores 1 and 2 begin execution from what is stored in the magic address. The L2_CONFIG input, read from the GPIO5 pin at bootload time, determines the magic address for DSP cores 1 and 2. When L2_CONFIG is 1, the magic address for both DSP core 1 and DSP core 2 is 8FFFFCh. When L2_CONFIG is 0, the magic address for DSP core 1 is 8FFFFCh and the magic address for DSP core 2 is 087FFFCh.

Table 16. Comparison of Bootloading Capabilities on the C6455 and C6474

Features		C6455	C6474
Number of Modes		6	11 (in secure and public versions)
Modes	No boot	Yes (1)	Yes (1)
	Host	Yes (1), HPI or PCI	No
	EMIF	8-bit (2)	No

(1) After boot, execution begins starting from 0800000h in L2 SRAM.
(2) After boot, execution begins starting from 0B0000000h in CE3 space in external memory.

Copyright © 2008, Texas Instruments Incorporated

Table 16. Comparison of Bootloading Capabilities on the C6455 and C6474 (continued)

Features		C6455	C6474
	I2C Master	Slave address = 50h	A) Slave address = 50h
			B) Slave address = 51h
	I2C Slave	Yes	Yes
	SRIO	Multiple configurations	Multiple Configurations
	EMAC	No	Master
			Slave
			Forced link
Selected by		BOOTMODE[3:0] pins	BOOTMODE[3:0] pins (GPIO[3:0])
		CFGGP[2:0] pins	CFG[3:0] pins (GPIO[11:8])
		PCI_EN pin	L2_CONFIG pin (GPIO5)

Table 16 shows a comparison of the bootloading capabilities of the C6455 and the C6474.

For detailed information regarding use of the bootloader on the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C6474 Bootloader User's Guide* (SPRUG24).

7 Power Management

In most DSP systems, power management is an important concern to allow DSP functions to be performed with the lowest power cost and minimal battery drain possible (for battery operated applications). The C6455 and the C6474 offer numerous options for power management.

Copyright © 2008, Texas Instruments Incorporated

On the C6455 and the C6474, there are several basic categories of power management options, and many different variations of these options within these basic categories. Both devices offer several options for managing power within the C64x+ DSP CPU core. These capabilities are implemented through the use of various specialized instructions and control registers within the DSP CPU, and control power within the CPU and internal memories. Both devices also provide the capability to enable or disable the main PLL (PLL1) under software control through the PLLCTL register.

On the C6455, additional power management capability is provided, primarily for the peripherals, through several mechanisms. Many of the peripherals on the device have external enabling pins, which serve to enable or disable the peripheral, and, when the peripheral is disabled, it is placed into a mode where its clocks and power are off, allowing its power dissipation to be reduced effectively to zero. The peripheral cannot be removed from this state through software, and, to be re-enabled, the enable pin must be switched, and a device reset performed. Additionally, if a peripheral is enabled with its external enabling pin, power to the peripheral can be controlled by enabling or disabling the clock to the peripheral through software control, using the Peripheral Configuration Registers (PERCFG0, PERCFG1). Finally, the EMAC RMII logic and trace output pin buffers can be enabled or disabled separately through software using their own dedicated control registers, the EMAC Configuration Register (EMACCFG) and the Emulation Buffer Powerdown Register (EMUBUFPD).

On the C6474, the main power management mechanism for the peripherals and functional modules within the device is provided through the use of the dedicated power and sleep controller (PSC) module, which implements the capability to turn on or off the clocks to peripheral and functional modules, therefore, controlling power usage on a module by module basis. The PSC is made up of the Global PSC (GPSC) module, which controls enabling power for the various peripherals and functional modules on the device, and the Local PSCs (LPSCs), which control enabling the clocks for each peripheral/module controlled. Through the use of the GPSC and the LPSCs, power usage on the C6474 can be optimized on the basis of actual module usage for a particular application.

The C6474 also implements SmartReflex™ silicon and circuit design power management, which is a feature that allows this device to automatically optimize its performance and power dissipation characteristics. For additional information regarding SmartReflex, see Section 10 of this document.

Table 17 summarize the power management options available on the C6455 and the C6474.

Table 17. C6455 and C6474 Power Management Options

Options	Controlled By	Controls
A	Specialized instructions/control registers	DSP CPU core + memory
B	PLL1 PLLCTL register	PLL1
C	External peripheral enable pins (1)	Peripheral modules
D	PERCFG0, PERCFG1 registers (1)	Peripheral modules
E	EMACCFG (1)	EMAC RMII logic
F	EMUBUFPD register (1)	Trace pins
G	GPSC/LPSC registers (2)	Individual peripherals/modules
H	SmartReflex (2)	Core V_{DD} voltage level

(1) Available on C6455 only.
(2) Available on C6474 only.

Copyright © 2008, Texas Instruments Incorporated

On the C6474, there are six power domains and 12 clock domains that are available to control clocks and power for the various peripheral/modules on the device. Table 18 and Table 19 present a listing of the power and clock domains on the C6474.

Table 18. TMS320C6474 Power Domains

Domain	Block(s)	Note	Power Connection
0 (AlwaysOn)	C64x+ Cores L1/L2 Memories Most peripheral logic I/Os	C64x+ Cores and L1/L2 can be placed in low power states.	C64x+ Cores and L1/L2 software control via C64x+ Core. Everything else isalways on.
1	AIF	RAMs can sleep, contents maintained	Software control
2	RapidIO	RAMs can sleep, contents maintained	Software control
3	RAC	RAMs can sleep, contents maintained	Software control
4	TCP	RAMs can sleep, contents maintained	Software control
5	VCP	RAMs can sleep, contents maintained	Software control

Table 19. TMS320C6474 Clock Domains

Module	Block(s)	Power Connection
0	C64x+ CORE 0 RSAs	Software control
1	C64x+ CORE 1 RSAs	Software control
2	C64x+ CORE 2 RSAs	Software control
3	C64x+ CORE 0	Always on
4	C64x+ CORE 1	Always on
5	C64x+ CORE 2	Always on
6	AIF	Software control
7	RapidIO	Software control
8	RAC	Software control
9	TCP	Software control
10	VCP	Software control
11	PSC, other peripherals, PLLs	Always on

For detailed information regarding power management on the C6474, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552), the *TMS320C6474 DSP Power/Sleep Controller (PSC) User's Guide* (SPRUG10), and the *TMS320C64x+ Megamodule Reference Guide* (SPRU871).

8 PLL/Clock Modes at Reset

Both the C6455 and C6474 feature flexible clock generators that provide clocking to satisfy a wide variety of system requirements. To properly start up and initialize a DSP system, the clock generator must be able to provide appropriate clock signals to the device even before the device is released from reset. Both the C6455 and C6474 clock generators are designed to provide this capability.

Both the C6455 and the C6474 require an external oscillator to provide clocks to the device (an internal oscillator is not provided); however, the C6474 uses differential clock inputs for increased noise immunity and signal integrity. See Section 4.11 for additional information regarding clock inputs on the C6474.

Copyright © 2008, Texas Instruments Incorporated

On the C6455, the default clock mode at reset for PLL1 is for the PLL to be bypassed; therefore, the input clock drives the internal clock generators directly. The input clock frequency range is 33.3 MHz to 66.6 MHz. Two of the clock dividers on the C6455 use fixed divider values, while D4 and D5 default to divide values of eight and four, respectively, at reset. The PLL mode and multiplier value, along the D4 and D5 divider values, can be changed through software after reset by writing to registers in the PLL module. Note that the bootloader may change clock mode and frequencies, depending on the bootload mode, once it starts running.

For PLL2 on the C6455, the input frequency range is 12.5 MHz to 26.7 MHz, and the default mode at reset is for the PLL to run at multiply-by-20, followed by a divide-by-two, resulting in an overall multiply-by-ten of the input clock. The D1 divider value can be programmed to either two or five, but defaults to two at reset.

On the C6474, the input clock frequency range for the main PLL (PLL1) is 40 MHz to 62.5 MHz, while the AIF must be clocked at either 61.44 MHz, 122.88 MHz, or 153.6 MHz. At reset, PLL1 is bypassed, and the internal clock generators run at the input clock frequency. The CORECLKSEL input pin determines whether PLL1 is driven by SYSCLKP/N or ALTCORECLKP/N. Six of the eight output dividers run a fixed divide ratios; the two programmable dividers, D11 and D13, default to divide-by-ten and divide-by-six, respectively, at reset. The PLL mode and multiplier value, along the D11 and D13 divider values can be changed through software after reset by writing to registers in the PLL module. Note that the bootloader may change clock mode and frequencies, depending on the bootload mode, once it starts running.

For PLL2 on the C6474, the input clock frequency range is 40 MHz to 66.7 MHz. PLL2 is always enabled, and runs at a fixed multiply-by-ten rate.

Table 20 presents a summary of the C6455 and C6474 PLL and clock mode initialization.

Table 20. C6455 and C6474 PLL/Clock Modes at Reset

Features	C6455	C6474
PLL1 Controller input clock frequencies	33.3 - 66.6 MHz	40 - 62.5 MHz
AIF input clock frequencies		61.44/122.8/153.6 MHz
On-chip oscillator	No	No
PLL1 mode at reset	Bypass	Bypass
Clock mode at reset determined by	Not variable (1)	CORECLKSEL pin ((1), (2))
Reset values of programmable dividers	D4: 8, D5: 4	D11: 10, D13: 6
Software can change clock rates after reset	Yes - some but not all	Yes - some but not all
PLL2 Controller input clock frequencies	12.5 - 26.7 MHz	40 - 66.7 MHz
On-chip oscillator	No	No
PLL2 mode at reset	x20, /2 = x10	x10
Clock mode at reset determined by	Not variable	Not variable
Reset value of programmable divider	D1: 2	
S/W can change clock rate after reset	Yes (3)	No

(1) Bootloader may change clock frequency after reset. See Section 6.
(2) Selects SYSCLKP/N or ALTCORECLKP/N to clock main PLL (PLL1).
(3) PLL parameters are fixed, but output divider can be changed.

For detailed information regarding PLL and clock mode initialization, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C6474 DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (SPRUG09).

9 Pin Multiplexing

The C6455 uses multiplexing of functions on various pins to allow multiple functions on the device to share the same external pin connections. The C6474 does not use pin multiplexing; therefore, there are no pin multiplexing considerations when migrating an application from the C6455 to the C6474.

Copyright © 2008, Texas Instruments Incorporated

10 Power Supply Considerations

The C6455 and the C6474 utilize multiple power supplies to maximize flexibility and performance and minimize power dissipation, as well as to adhere to industry standards for various external interfaces. Additionally, some of the different clock speed versions of the devices have different power supply voltage level requirements.

The C6455 utilizes a 1.2 V or 1.25 V supply (depending on speed) for the DSP core CPU and internal logic, 3.3 V for some of the I/O pins, and a 1.8 V power supply for the DDR2 interface and some of the other I/O pins. In addition, the EMAC can use either a 1.8 V power supply, or, optionally, a 1.5 V supply.

On the C6474, the core power supply voltage is managed utilizing SmartReflex, a feature that optimizes power dissipation and speed based on process variation. With SmartReflex, the C6474 device is tested during manufacturing and programmed with a code specifying the optimal core power supply voltage. Then, during operation, a programmable voltage regulator is used to provide the core power supply with the optimal voltage based on this code. In this way, the C6474 device can function at its maximum speed while still operating within its specified power dissipation. This allows the C6474 to effectively provide three times the performance of the C6455 at 3.4MIPS/mW (compared to the C6455's 2.9 MIPS/mW) while dissipating less than three times the power (7 watts for one C6474 compared to 9.9 watts for three C6455s) when executing a typical application.

Two other power supplies are required on the C6474: a 1.8 V power supply is required for DDR2 and some of the I/O and a 1.1 V power supply is required for the other I/O pins, along with the SERDES modules associated with the AIF, EMAC and SRIO.

Table 21 summarizes the power supply requirements for the C6455 and the C6474.

Table 21. Power Supply Requirements

Supply		C6455		C6474
Core	Speeds	1 GHz/1.2 GHz	720 MHz/850 MHz	All
		1.25 V	1.20 V	0.9 V - 1.2 V (SmartReflex)
I/O		3.3 V		
I/O (DDR2, others)		1.8 V		1.8 V
I/O (EMAC, optional)		1.5 V		
I/O (SERDES)				1.1 V

For additional detailed information regarding power supply requirements on the C6474 including SmartReflex, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552) and the *TMS320C6474 Hardware Design Guide* (SPRAAW7).

11 Package and Pin Count Comparisons

The C6455 and the C6474 are both provided in cost-efficient, high density BGA packages. Since the two devices have different pinouts, pin connections and locations are different between the two devices; therefore, PC board layout and signal connection modifications are necessary when migrating from a C6455 to a C6474 device. Note that both devices are available in lead-free and non-lead-free packages. Also note that the C6455 is available in both commercial and extended temperature ranges.

Copyright © 2008, Texas Instruments Incorporated

Table 22 shows a comparison of the packages and pin counts for the C6455 and the C6474.

Table 22. Package and Pin Count Comparison

Characteristic	C6455	C6474
Types	697 pin ZTZ (lead free) 697 pin GTZ (not lead free)	561 pin CUN (lead free) 561 pin GUN (not lead free) 561 pin ZUN (not lead free)
Ball Pitch	0.8 mm	0.8 mm
Dimensions	24 x 24 mm	23 x 23 mm
Max Case Temp	90°C (commercial) 105°C (extended)	100°C (commercial)

For further detailed information regarding pinout, mechanical dimensions, thermal characteristics, and lead content of the C6474 packages, see the *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552).

12 References

- *TMS320C6474 Multicore Digital Signal Processor Data Manual* (SPRS552)
- *TMS320C64x+ DSP Cache User's Guide* (SPRU862)
- *TMS320C6474 Antenna Interface User's Guide* (SPRUG12)
- *TMS320C6474 Frame Synchronization User's Guide* (SPRUG13)
- *TMS320C6474 DSP Viterbi-Decoder Coprocessor 2 (VCP2) Reference Guide* (SPRUG20)
- *TMS3206474 DSP Turbo-Decoder Coprocessor 2 (TCP2) Reference Guide* (SPRUG21)
- *TMS320C6474 DDR2 Implementation Guidelines* (SPRAAW8)
- *TMS320C6474 DSP DDR2 Memory Controller User's Guide* (SPRUG19)
- *TMS320C6474 DSP Enhanced DMA (EDMA3) Controller User's Guide* (SPRUG11)
- *TMS320C6474 Serial RapidIO (SRIO) User's Guide* (SPRUG23)
- *TMS320C6474 SERDES Implementation Guidelines* (SPRAAW9)
- *TMS320C6474 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide* (SPRUG17)
- *TMS320C6474 DSP Software-Programmable Phase-Locked Loop (PLL) Controller User's Guide* (SPRUG09)
- *TMS320C6474 Hardware Design Guide* (SPRAAW7)
- *TMS320C6474 DSP 64-Bit Timer User's Guide* (SPRUG18)
- *TMS320C6474 DSP Ethernet Media Access Controller (EMAC)/Management Data Input/Output (MDIO) User's Guide* (SPRUG08)
- *TMS320C6474 DSP Inter-Integrated Circuit (I2C) Module User's Guide* (SPRUG22)
- *TMS320C6474 Semaphore User's Guide* (SPRUG14)
- *TMS320C6474 DSP General-Purpose Input/Output (GPIO) User's Guide* (SPRUG16)
- *TMS320C64x+ DSP Megamodule Reference Guide* (SPRU871)
- *TMS320C6474 DSP Chip Interrupt Controller (CIC) User's Guide* (SPRUFK6)
- *TMS320C64x/C64x+ DSP CPU and Instruction Set Reference Guide* (SPRU732)
- *TMS320C6474 Bootloader User's Guide* (SPRUG24)
- *TMS320C64x+ Megamodule Reference Guide* (SPRU871)
- *TMS320C6474 DSP Power/Sleep Controller (PSC) User's Guide* (SPRUG10)

Copyright © 2008, Texas Instruments Incorporated

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated